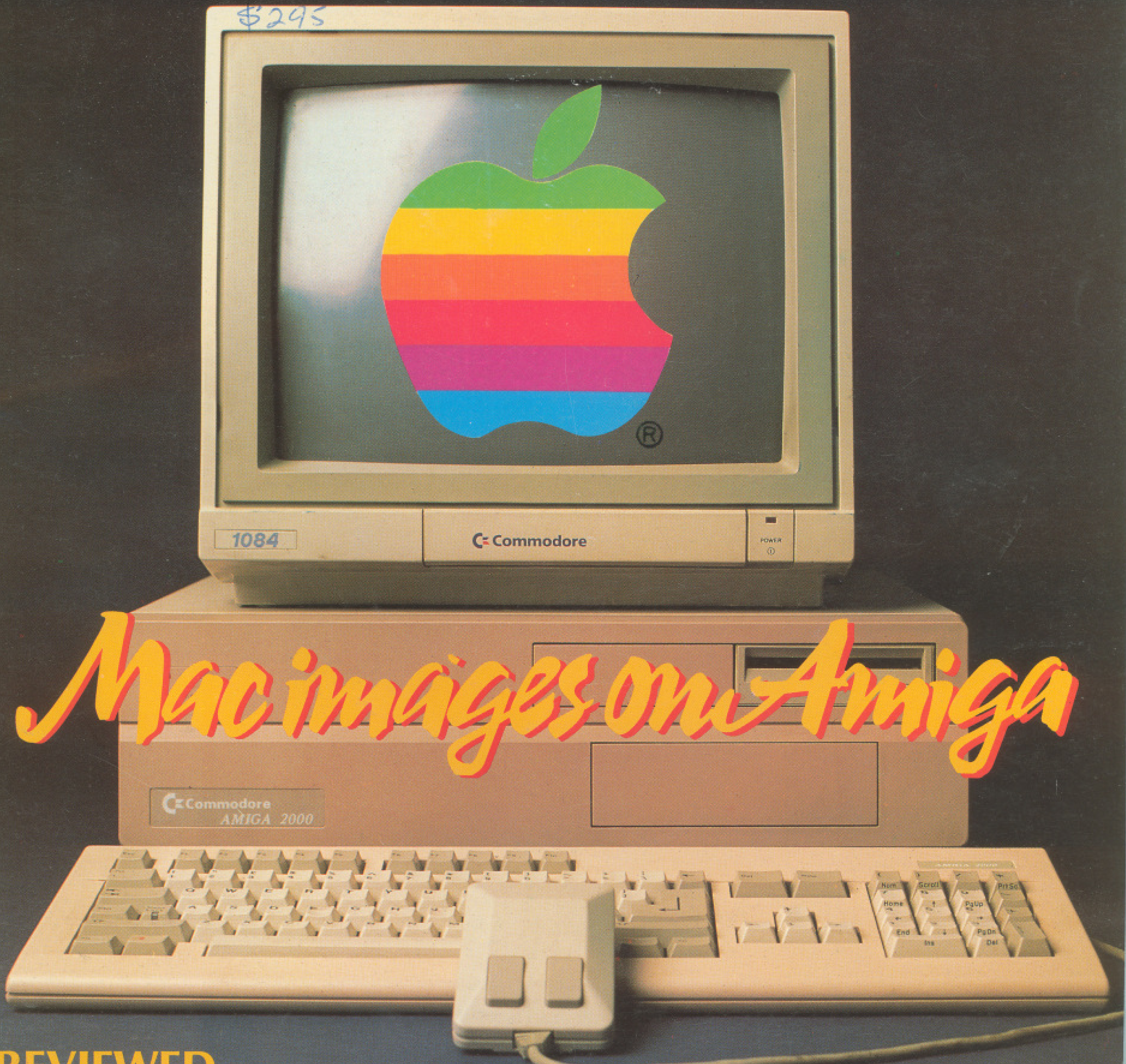
PLUS 4 · AMIGA · PC · C64 · C128 · C16 PLUS 4 · AMIGA

AUGUST 1989 £1.30

YOUR COMMODORE



Mac images on Amiga

REVIEWED

On The PC: 688 Attack Sub, Abram's Battle Tank, Battle Hawks 1942

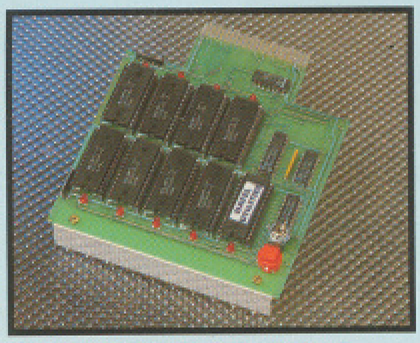
For The C64: Project Firestart, Speedball, Navy Moves, Star Trek

UNBEATABLE PROGRAMS

Address Book, Disk Scrambler, Thesauras, Window on a maze

ISSN 0269-8277
08
9 770269 827007

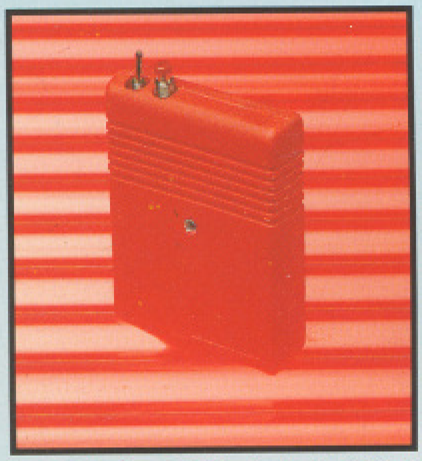
DATEL ELECTRONICS



256K SUPEROM EXPANDER

- Select instantly from 8 sockets which accept up to 32K EPROM each.
- Program your own EPROMs using our EPROM programer.
- No need to have lots of cartridges - just make a selection from the Superom menu.
- Fully menu driven on power up.
- Select any slot under software controls.
- Unique EPROM generator feature will take your own programs - basic or m/c & turn them into autostart EPROMs. (EPROM burner required).
- Accepts 2764/27128/27256 EPROMs.
- On board unstoppable reset.
- On board operating systems - no programs to load.

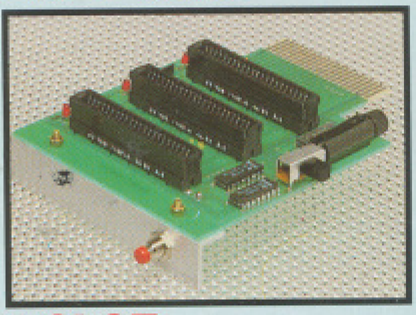
ONLY £29.99



RESET CARTRIDGE

- Unstoppable reset button. NOTE:- user port cheaper type reset buttons offered by others are not unstoppable.
- Resets even so called "unstoppable" programs.
- Add pokes from magazines etc.
- Fits in cartridge port.
- Simply plug in.

ONLY £5.99

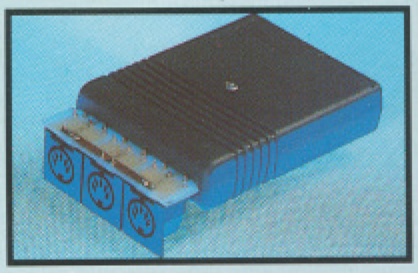


3 SLOT MOTHERBOARD

SAVE WEAR & TEAR ON YOUR EXPANSION PORT

- Accepts 3 cartridges.
- Switch in/out any slot.
- Onboard safety fuse.
- Fully buffered.
- High grade PCB.
- Reset button.

ONLY £16.99



MIDI 64

FULL MIDI INTERFACE FOR THE 64/128 AT A REALISTIC PRICE.

- MIDI in/MIDI thru/2 x MIDI out.
- Compatible with most leading software packages.

ONLY £29.99



DIGITAL SOUND SAMPLER

- The new sampler allows you to record any sound digitally into memory & then replay it with astounding effects.
- Playback forwards/backwards with echo/reverb/ring modulation.
- Now with full sound editing module to produce outstanding effects.
- Full 8 bit D to A & ADC conversion.
- MIDI compatible with suitable interface. (i.e. Datel unit for £29.99, see ad.).
- Live effects menu includes real time display of waveforms.
- Line in/mic in/line out/feedback controls.
- Powerfull sequencer with editing features.
- Load/save sample.
- Up to 8 samples in memory at one time.
- Tape or disk (please state).
- Complete software/hardware package.

ONLY £49.99

COM-DRUM

DIGITAL DRUM SYSTEM

- Now you can turn your digital sound sampler into a digital drum system.
- 8 digital drum sounds in memory at one time.
- Complete with 3 drum kits.
- Real drum sounds - not synthesised.
- Create superb drum rhythms with real & step time.
- Full editing. Menu driven.
- Output to hi-fi or through TV speaker.
- Load/save facilities.

ONLY £9.99

STATE TAPE OR DISK

COM-DRUM EDITOR

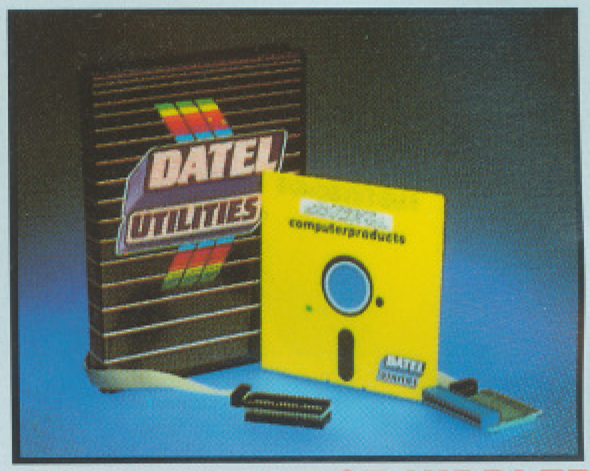
- 24 drum sounds supplied on disk to enable you to construct your own drum kit.
- Re-arrange sounds supplied with a Com-Drum to make new kits.
- Load/save facilities.

ONLY £4.99 DISK ONLY

DATA RECORDER

- Quality Commodore compatible data recorder.
- Pause control, counter, etc.
- Suitable for 64/128.
- Send now for quick delivery.

ONLY £24.99



DEEP SCAN BURST NIBBLER™

- The most powerfull disk nibbler available anywhere, at any price! Burst Nibbler is actually a two part system - a software package & a parallel cable to connect the 1541/1570/1571 to 64/128 (state type).
- What gives Burst Nibbler it's power? Conventional nibblers have to decode the data from the disk before it can transfer it using the serial bus - when non standard data is encountered they are beat. Burst Nibbler transfers data as raw GCR code via the parallel cable without the need to decode it so you get a perfect copy of the original.
- • Will nibble up to 41 tracks • Copy a whole disk in under 2 minutes • Full instructions • Regular updates - we always ship the latest • Fitted in minutes - no soldering usually required • Full or 1/2 tracks • No need to buy parallel cable if you have Disk Demon/Dolphin etc. • Cable has throughbus extension for other add ons.
- Wether to choose FastHack'em or Burst Nibbler? Fast Hack'em is unbeatable value as an "all rounder" - with nibblers, 1 or 2 drive copy, format, file copy, 1571 copy etc. etc., so if you have a more general requirement perhaps Fast Hack'em is for you. Burst Nibbler is a pure nibbler second to none, for the reasons stated. So if it's just making backups you are interested in, there is no other product to beat it!

ONLY £24.99 COMPLETE

SOFTWARE ONLY £12.99

CABLE ONLY £14.99



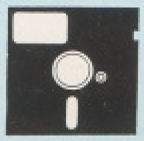
ROBOTEK 64

MODEL & ROBOT CONTROL MADE EASY.

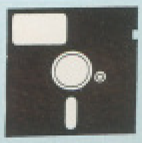
- 4 output channels - each with onboard relay.
- 4 input channels - each fully buffered TTL level sensing.
- Analogue input with full 8 bit conversion.
- Voice input for voice control.
- Software features:- test mode/analogue measurement/voice activate/digital readout etc.

ONLY £39.99

INCLUDING HARDWARE/SOFTWARE/MIC. ETC,(STATE TAPE OR DISK).



TOOLKIT IV

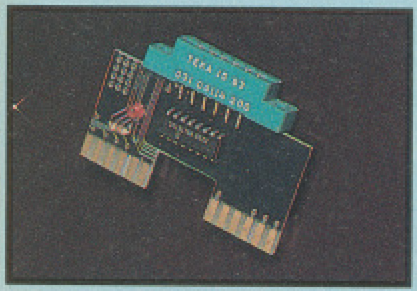


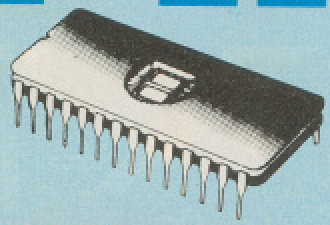
THE ULTIMATE DISK TOOLKIT FOR THE 1540/1541.

- A disk toolkit is an absolute must for the serious disk hacker. Toolkit IV has more features than most for less.
- DISC DOCTOR V2 - Read & write any track & sector including extra & renumbered tracks. Repair damaged sectors. Look underneath read errors.
- HEADER/GAP EDITOR - Decodes & displays ALL header information including off bytes & header gap. Rewrite the entire header & header gap. Renumber sectors. Also edit any sector tail gap.
- DISK LOOK - Sort directory. Recover lost files. Display file start /end addresses. Disassemble any file program directly from the disk to SCREEN or PRINTER including undocumented opcodes. Edit Bam.
- FAST FILE COPY - Selective file copy. Works at up to 6 times normal speed.
- FAST DISK COPY - Copy an entire disk in 2 minutes or less using single 1541.
- FILE COMPACTOR - Can compact machine programs by up to 50%. Save disk space. Compacted programs run as normal.
- FORMATTER - 10 second format an entire disk or format any individual track or half track 0 to 41. Re-define any of 30 parameters to create or recreate unique disk formats.
- ERROR EDIT - Quickly find & recreate all read errors including extras & renumbered tracks or sectors & half tracks from 0 to 41. Even recreates data under errors & allows you to redefine any necessary parameters.

ONLY £9.99

DATEL ELECTRONICS





NEW CLONEMASTER

- The most effective tape to tape back-up board available.
- Makes perfect backups of your tapes easily & effectively!!
- No user knowledge needed at all.
- On board TTL logic circuitry actually shapes the program & sends a perfect signal to the record cassette - producing a copy better than the original in many cases.
- L.E.D. indicator shows when data is being transferred to avoid exessive tape winding.
- Works with almost any program including multi-loaders, turbos & even very unusual type turbos.
- Requires access to two CBM compatible data recorders.
- Simply press 'Play' on one recorder & press 'Record' on the other - that's it!
- You can even make a backup while you are loading the program.
- This is a total hardware solution - no programs to load - the results are stunning!

ONLY £9.99

TURBO ROM II

TURBO ROM II IS A REPLACEMENT FOR THE ACTUAL KERNAL INSIDE YOUR 64. IT PROVIDES SUPERFAST LOAD/SAVE ROUTINES.

- Loads most programs at 5-6 times normal speed.
- Saves at 5-6 times normal speed as well.
- Improved DOS support including 10 sec. format.
- Programed function keys:- load, directory, old, etc.
- Return to normal kernal at a flick of a switch.
- FCOPY-250 block file copier.
- FLOAD-special I/O loader.
- Plus lots more.
- Fitted in minutes - no soldering usuallly required. (On some 64's the old ROM may have to be desoldered).

ONLY £14.99



SMART CART

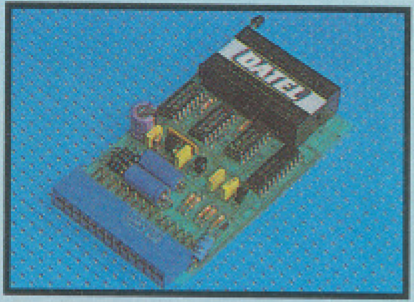
- 32K pseudo ROM.
- Lithium battery lasts up to 5 years.
- Simply load the program you require - then flick the switch. The cartridge can then be removed just like a ROM cartridge.
- Make your own cartridges including autostart types - without EPROM burner. 32K version = 4 x 8K pages.
- Some knowledge of M/C is helpful - but full instructions included.
- I/O 2 slot open for special programing techniques.

ONLY £29.99

RAM DISK

- Turn your Smart Cart into a 32K Ram/disk.
- 32K of instant storage area for files/programs.
- Disk type commands:- load, save, directory, scratch.
- Program data retained when computer is switched off!
- Full command set with instructions.

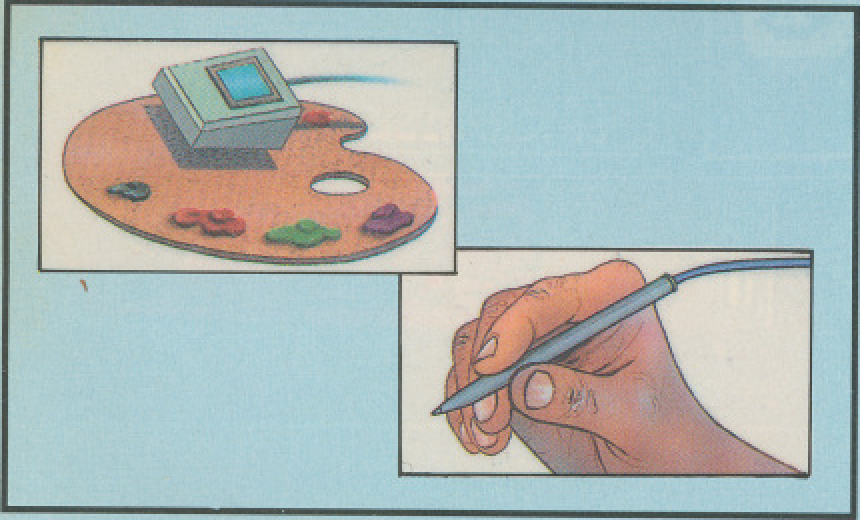
ONLY £9.99



EPROMMER 64™

- A top quality, easy-to-use EPROM programer for the 64/128.
- Fully menu driven software/ hardware package makes programing/reading/verifying/ copying EPROMs simplicity itself.
- Will program 2716, 2764, 27128 & 27256 chips. 12.5, 21 or 25 volts.
- Fits into user port for maximum compatibility with cartridges/ Superom Board etc.
- Full feature system - all functions covered like device check/verify.
- We believe Eprommer 64 is the most comprehensive, most friendly & best value for money programer available for the 64/128.
- Ideal companion for Superom Board, Cartridge Development System, our kernal expanders or indeed any EPROM base project.
- Comes complete with instructions - plus the cartridge handbook.

ONLY £39.99
COMPLETE



BLAZING PADDLES

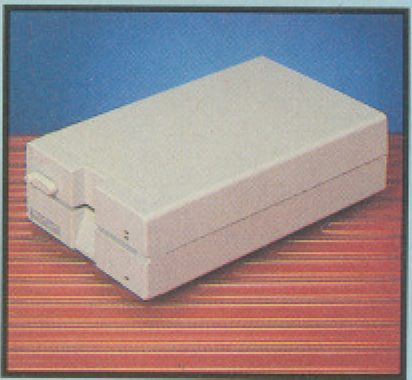
A COMPLETE LIGHTPEN/GRAPHICS ILLUSTRATOR PACKAGE.

- A fully Icon driven graphics package of a calibre which should cost much, much more.
- Complete with a fibre optical lightpen system for pin point accuracy.
- Pictures can be saved from 'Action Replay' & edited with Blazing Paddles.
- Blazing Paddles will also work with many other input devices including:- Joysticks, Mice, Graphics Tablets, Trackball, etc.
- Multi feature software including:- • Range of brushes • Airbrush • Rectangle • Circle • Rubberbanding • Lines • Freehand • Zoom mode • Printer dump • Load/save • Advanced colour mixing - over 200 hues! • Cut & paste allows shapes/windows/pictures to be saved to/from tape/disk.

ONLY £24.99
COMPLETE BLAZING PADDLES & LIGHTPEN

BLAZING PADDLES ONLY
£12.99

BLUE CHIP DRIVE



- External power supply for cool operation.
- Even works in 1571 double sided mode (128 mode).
- Fully C64/128 compatible.
- This drive is now probably the most compatible drive available for the Commodore. More so than even Commodores own '1541C'.
- Supplied complete with all cables - no more to buy.
- At last, a top quality drive at a sensible price.

ONLY £129.99
ADD £5.00 FOR COURIER DELIVERY

ULTRA CRUNCHER

- The ultimate cartridge based program compactor.
- Compacts by up to 30%!
- 3 compacting programs on one cartridge.
- Fast loading/saving routines.
- Full DOS support including fast format.

ONLY £12.99
on cartridge

Available as chip only for 256K Superom Board fcr
ONLY £7.99

FAST HACK'EM™

- Multi Module Disk Nibbler - all on the one disk.
- Single 1541 Nibbler - Copy an entire disk in just 2 minutes.
- Super fast File Copy - typically copies an average file in under 9 seconds.
- No other product is such a good "All Rounder" at this incredibly low price...

ONLY £9.99

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0782 744707 24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	0782 744292 UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLERS WELCOME - Please reserve goods by telephone prior to visit.

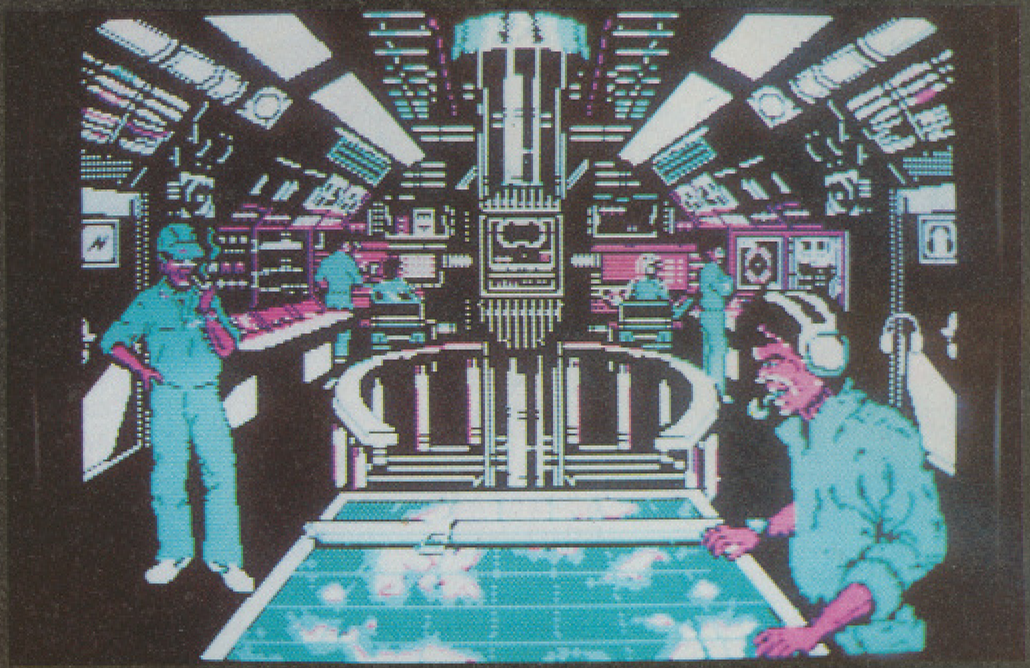
DATEL ELECTRONICS

DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.

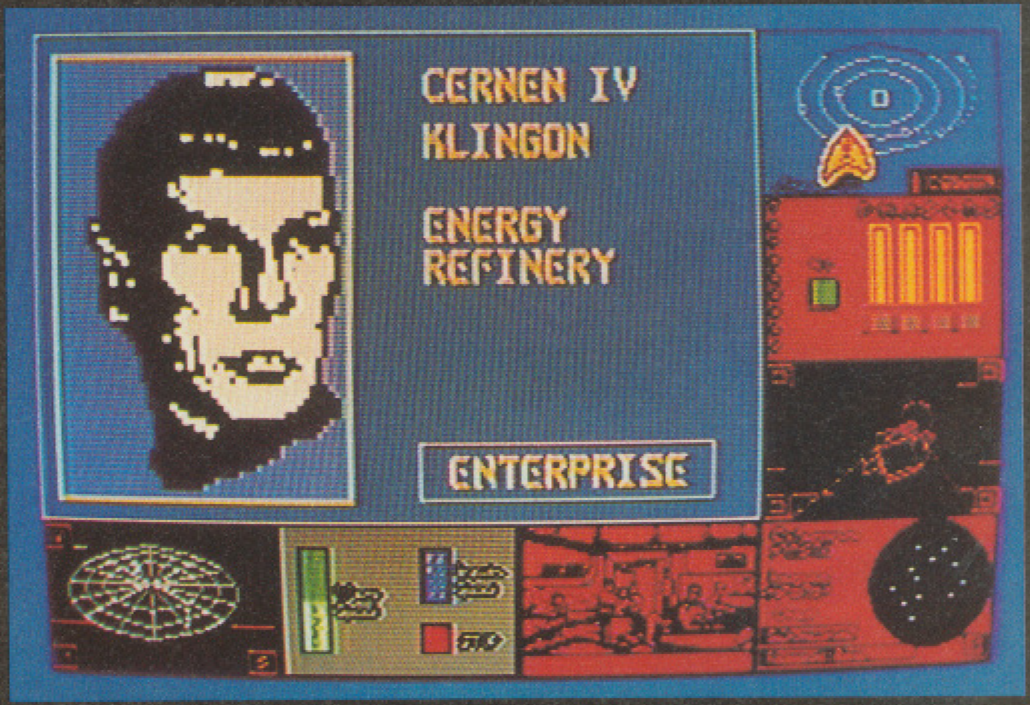
SALES ONLY	TECHNICAL ONLY
0782 744707	0782 744324

CONTENTS

VOLUME 5
NUMBER 11



PC Games



Startrek



Macpics

FEATURES

REGULARS

PROGRAMS

FourGround!

Plus/4 Past and Present...

By Mark Everingham

1984 – the year of the first untethered spacewalk, Carl Lewis takes four gold medals in the Summer Olympics, Desmon Tutu wins the Nobel Peace Prize, and Commodore introduce the Plus/4 computer. All right, so maybe the advent of the Plus/4 did not have *quite* the same impact, but its arrival did create something of a stir in the microcomputer world. The magazine *Your Computer* described the machine as a "QL-Basher" and suggested that the Plus/4 could replace the BBC Micro in school and business because of the powerful CBM BASIC V3.5 and the integrated business software. Unfortunately, these ambitious claims were not to be fulfilled. In recent times the Plus/4 has lost the support of most software houses, and has sadly slipped into relative obscurity. Even so, the Plus/4 did have considerable, if short-lived success in both the UK and, perhaps more importantly, in Germany and the USA. I was convinced myself of how many people are still using the machine by responses to a bug which appeared in a program of mine published in *Your Commodore* last year. I received numerous letters from as far as the States and Holland, and was impressed by how many people had actually had a go at solving the bug, rather than justing writing for help. Although largely forgotten by the software industry, the Plus/4 is still far from dead, five years after its conception, and now it has finally received the recognition it deserved, in this form – an area of the magazine set aside soley for the Plus/4 computer, and the name – **FourGround!**

Aims of The Series

When I was approached by *Your Commodore* to write this series on the Plus/4 I accepted immediately – the Plus/4 is simply the most friendly, useable machine I have ever come across and has been disregarded for far too long. However, when I started to think of some ideas to write about, I realized the enormous job I had taken on. The fact is that the Plus/4 excels in so many things, and has so many different uses from business to games playing that it is hard to target any one area of interest. If I confine myself to machine language programming I may alienate the Basic-only programmers, and of course I mustn't forget those who use their Plus/4 mainly for playing games. I have therefore selected a number of aspects of the Plus/4 which I consider to be definitive of the computer, and intend to cater for as many diverse interests as possible. If demand is high enough, I would also be happy to run a "Plus/4 Problems" type section. If you have any ideas, suggestions, questions or requests, please write to me at the address shown at the end of the article. If you require a reply, please do include an S.A.E.

Having got the introductions over, I thought that this month rather than leaping straight into a new topic, we'd take a look at what the Plus/4 offers. What follows is a brief overview of the Plus/4 system, and an inventory of those features of the Plus/4 which will be covered in this space in future issues of the magazine. Because the Plus/4 has so much to offer, the less obvious aspects of the computer are often forgotten. This month's overview should act as a memory refresher and a taster of subjects for future coverage.

An Overview Of The Plus/4 Computer

The Plus/4 computer has essentially three modes of operation – running Basic programs, running the built-in business software, and running commercial or the user's own machine-code programs. The dialect of Basic provided with the machine is not the C64's old Basic V2.0 which goes back to CBM PET and VIC-20 days, but is a far improved version 3.5. The main difference is the provision of commands to handle graphics and sound, which had to be performed using endless strings of POKEs on the Plus/4's big brother, the C64. In addition, it is possible to write far more elegant and structured programs in Basic 3.5 because of the **DO...LOOP** structures omitted in the earlier versions of Commodore Basic. The set of graphics commands included are as complete a collection as can be found on *any* computer. Basic 3.5 also offers comprehensive "house-keeping" facilities like program renumbering and automatic line numbering which along with a set of eight redefinable function keys make programming in Basic simplicity itself. Commodore's enhanced full-screen editor found on the Plus/4 means that prototyping of screen displays is easy, and program editing fast and reliable. Although Plus/4 Basic is not the speediest language known, its importance and usefulness should not be underestimated. By its use of ROM/RAM paging, Basic can use the full 64k RAM of the Plus/4 computer. However, arguably the most useful aspect of Basic 3.5 is that it offers unique routines which allow the machine-code programmer to easily add new commands to the Basic language. This means that there should never be any need to use unfriendly **POKEs** and **SYS** calls. Unfortunately, like so many of the Plus/4's other features, the extension of Basic does not seem to be documented in any books or manuals, so you can expect a full expose in the coming months.

The second major mode of Plus/4 operation is its built-in "3+1" business software: the integrated word-processor, spreadsheet, database and graph software. Too often this aspect of the Plus/4 is totally disregarded, yet while the software is a bit spartan, it *is* useable and with a little ingenuity it can be made to perform a wide range of extra functions due to Commodore's generous use of RAM-vectors. I shall be presenting in **FourGround** programs which overcome the lack of word-count, copy quantity when printing, apostrophe conversion as well as providing more complex functions such as control-code handling and facilities for the insertion of screen pictures into documents just like the graphical word-processors available on the Amiga.

By far the most interesting and powerful aspect of the Plus/4 is machine-code, otherwise known as *Machine Language* or *Assembly Language*. Because of the Plus/4's built-in machine-code monitor *TEDMON*, the Plus/4 is the perfect machine on which to learn machine language. The 7501 Processor in the Plus/4 is easy to learn and powerful in its simplicity. The inclusion of a Kernel ROM in the Plus/4 makes device handling, disk-drive access and printer control easy to grasp. Machine language regretta-

bly has a certain stigma attached to it, having gained the reputation of a language beyond the reach of normal mortals. This is *not* the case – machine-code can be simple and rewarding to use. The problem is usually a bad method of teaching, so I shall be presenting a readable, easily understood introduction to this fascinating aspect of Plus/4 computing some time in the future.

Of course, possibly the most important aspect of Plus/4 machine-code is correct use of the TED chip which replaces the VIC chips found in the older C64 and VIC-20 machines. The TED chip is a complex graphics/sound handling chip which offers a host of features:

High-resolution Text; Multi-colour Text and Extended Mode screen displays; 121 colours and redefinable character-sets of either 128 or 256 characters; Hardware controlled flashing and text reversal; High-resolution and Multi-color graphics modes with facilities for split-screen operation; Hardware smooth scrolling in both text and graphics modes; Raster (Screen Position) interrupts; Timer interrupts at speeds of up to 1Mhz and three hardware timers; one reloadable; Full ROM/RAM paging and facilities for paging internal 32k Firmware ROM chips; Two channel sound output and white noise generator... The list could go on for ever. Thre are simply so many features that while most people understand the basic principles, they do not appreciate just how versatile even the simplest funciton can be when used in unconventional ways. As an example, try entering this listing.

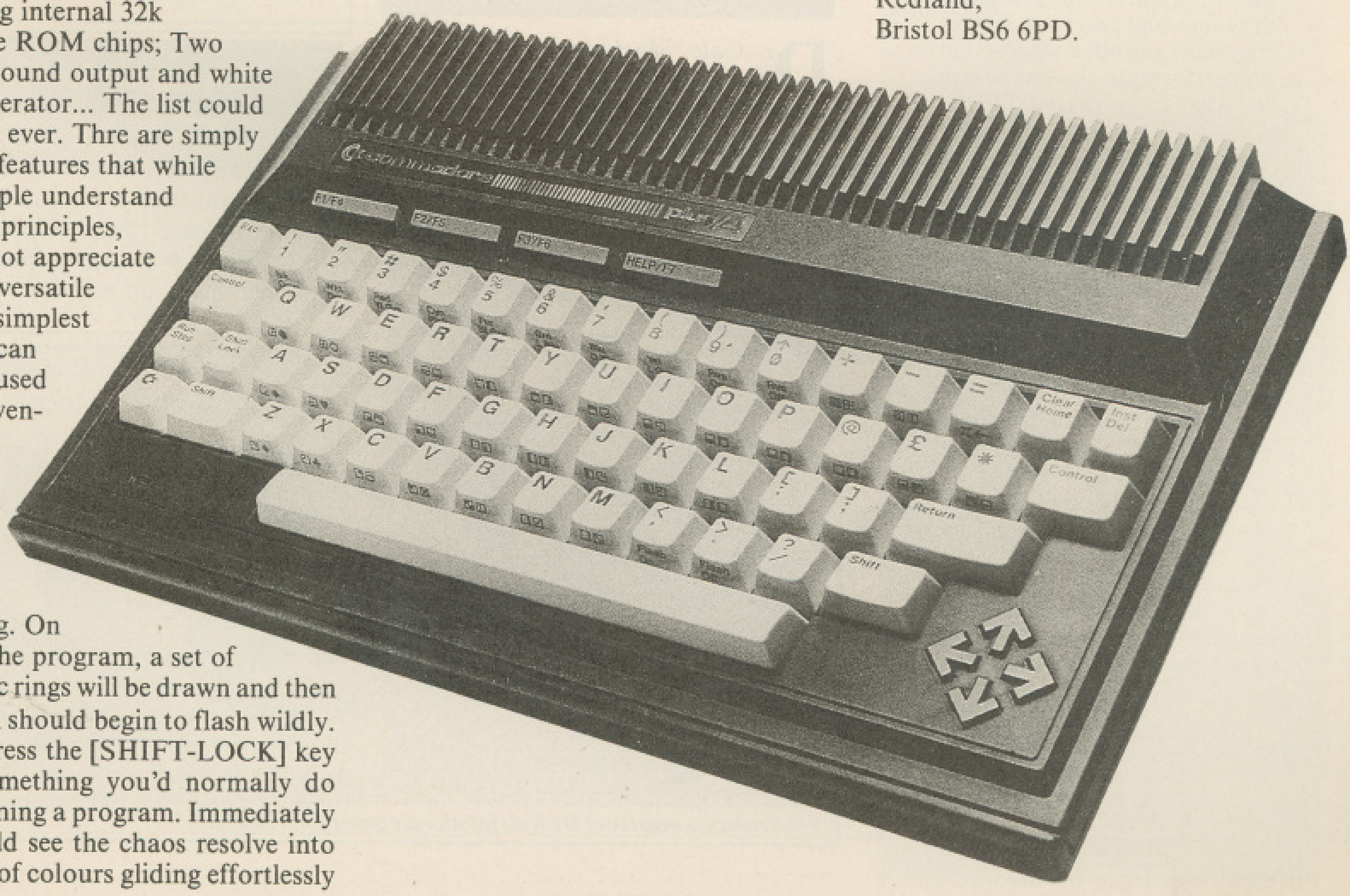
```
10   COLOR 0,2:COLOR 4,1,0:
     COLOR 1,1,0
20   GRAPHIC 1,1:P=1
30   FOR R=0 TO 90 STEP 10
40   CIRCLE 1,160,100,R:P=-P
50   IF P=1 THEN PAINT
     1,160,101-R
60   NEXT R
70   COLOR 0,1,0
80   GRAPHIC 1:GRAPHIC 3
90   GOTO 80
```

On running the program, a set of concentric rings will be drawn and then the screen should begin to flash wildly. Now depress the [SHIFT-LOCK] key – Not something you'd normally do when running a program. Immediately you should see the chaos resolve into a pattern of colours gliding effortlessly up the background of the screen; amazing when you consider that the whole special effect is being produced by just three Basic commands and not a spot of machine-code in sight. So how does the program work? If you stop the program and type **GRAPHIC 1** you'll see a pattern of black and white rings, but in **GRAPHIC 3**, the same pattern becomes different colours in the Multi-color mode. Normally, switching between the two screens just causes the display to flicker. Depressing the [SHIFT-LOCK] key, or holding down any other key makes the Plus/4's interrupts run a bit slower as the keyboard is scanned. This in turn brings the speed of flicker down to a rate close to the TV picture updating speed, resulting in the rolling colours effect. OK, so it's really just a gimmick, but from the obscure use of a simple function, we've learnt something about graphics modes, keyboard scanning, interrupt handling and TV Picture Handling (or the ubiquitous Raster). It is often unconventional little programs like this which teach you the most. So, if you have any interesting short programs, send them in!

To complement such Plus/4-specific subjects, I should like to include features on computing previously not tried on the Plus/4, but fully within its capabilities: Simple artificial intelligence and strategy programming for example. Another subject that you can be sure of coverage in the near future is the mandelbrot set and the whole field of fractal and chaotic maths. This exhilarating new area of computing has usually been confined to the Amigas and PCs of the world, but some beautiful effects can be achieved with incredibly simple Basic programs on the Plus/4, with its 121 colours adding a whole new dimension to the subject.

The Plus/4, as I have tried to demonstrate by this brief overview has a wide scope of ability for any interests. I hope I have whetted your appetite for things to come. Next month we'll kick off with something a bit more substantial. Remember, this section of the magazine should be for *you* the Plus/4 Owners, so please do write in! The address for any letters is:-

FOURGROUND
17 Collingwood Road,
Redland,
Bristol BS6 6PD.



Editor: *Stuart Cooke*
Deputy Editor: *Paul Whitington*
Technical Editor: *Paul Eves*
Group Editor: *Mark Webb*
Advertisement Manager: *Paul Kavanagh*
Ad-Copy Control: *Ruth Taylor*
Artist: *Alan Batchelor*
Designer: *Mark Newton*
Origination: *Ebony*
Printers: *Chase Webb*

Your Commodore incorporating Your 64 is a monthly magazine appearing on the first Friday of each month. Argus Specialist Publications Limited. Editorial and Advertisement office, *Your Commodore*, Argus House, Boundary Way, Hemel Hempstead HP2 7ST. Telephone: (0442) 66551. Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, 5 River Park Estate, Berkhamsted, Herts HP4 1HL. U.S.A. Subscrpition Agent: Wise Owl Worldwide Publications, 4314 West 238th Street, Torrance CA 90505 U.S.A.

ARGUS PRESS GROUP

ISSN 0269-8277

ABC

Data Statements

PD For All

Some of the best software available for the Commodore range of computers is available in the Public Domain. This is software that is freely distributable, as long as only a minimal charge is made for the disk and handling.

Kingsway Computer Services has informed us of the availability of its free PD catalogue, showing just what software the company is offering. Programs are availble on disk only, and cover a large range of subjects such as Education, Utilities, Home Accounts and Games. PD disks are available for the Plus/4, C64 and C128.

For your FREE Kingsway PD catalogue either write or telephone Kingsway Computer Services at 140 Rushdale Road, Sheffield, S8 9QE. Tel: (0742) 588429.

Over the Top?

De Gale Marking are launching four new joysticks, including a top-of-the-range stick that boldly goes where no joystick has gone before by giving the games player not only six fire buttons and an autofire with speed control, but also a built-in digital stopwatch. The three joysticks range in price from £8.95 to £19.95.

For Book Worms

Probably the best known UK publisher of Electronics and Computer books in the UK is Bernard Babani. A selection of the company's massive range of small paperback books can be found in most good book shops. Well, you may like to know that you can now obtain a free copy of their 1989 catalogue, giving information on all of their titles, ranging from *Secrets of the Commodore 64* to the *Electronic Calculator User's Handbook*.

For your free catalogue write to: Bernard Babani (publishing) Ltd. The Grampians, Shepherds Bush Road, London W6 7NF. Tel: 01 – 603 2581.

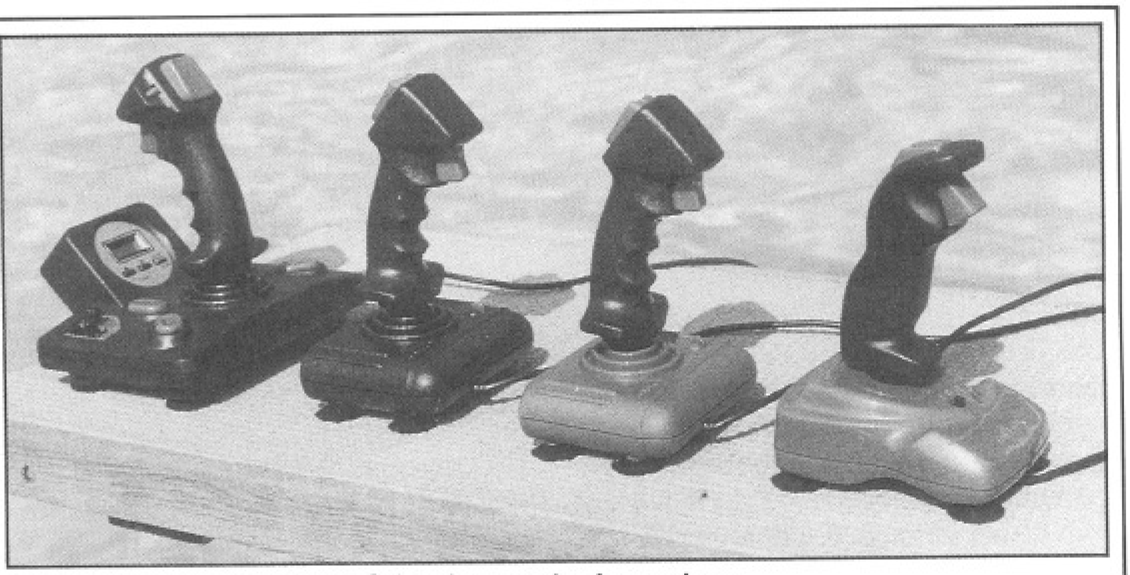
New Names for Imageworks

Anthony Taglione and Pete James, better known as Starlight Software's authors of the 8-bit games *Red L.E.D.* and *Deathscape*, have found a new home with Imageworks. *Bloodwych*, the team's first 16-bit game, should be out any time now on the Amiga, and a C64 version will soon follow.

Bloodwych is a fantasy role-playing game, but the horizontally split screen which offers a two player mode adds a new dimension to this genre of game.

Cheaper Lasers

Buying a Laser printer is normally an expensive business, but Qume has just launched a unique upgrading scheme to ease the purchase of their CrystalPrint WP and Series II printers. The user can buy a CrystalPrint



Yes, readers, a quartet of De Gale joystics are going for a swim...

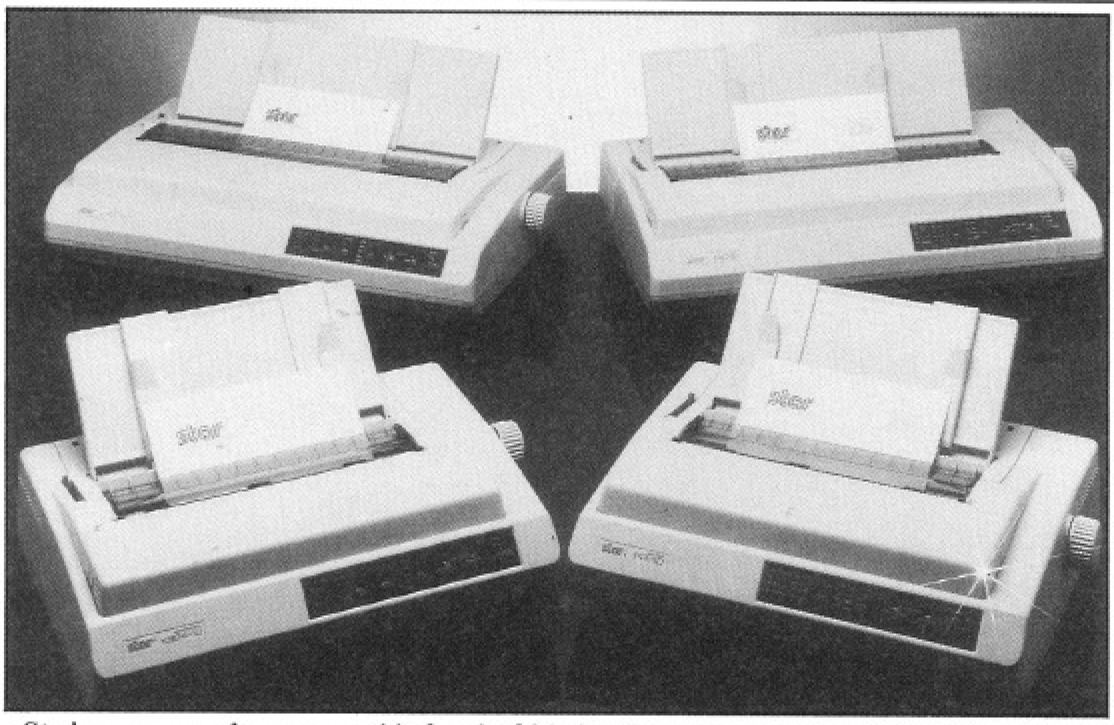
WP at £995 for basic text editing, and upgrade it to a series II when text and graphics are required. The series II (£1,495) and the WP can be modified to the top of the range Publisher PostScript language compatible, which retails at £2,999.

Bunk Desks

NDS Industries seem to have found a solution to overcrowded offices with the introduction of a new two-tier workstation. The two tier system is designed for use in computer rooms where space is at a premium, and there are a large number of terminals around.



Taglione and James, authors of Bloodwych



Star's new range of very competitively priced 24-pin printers

Seeing Stars On Site

If you require high quality output from your dot matrix printer, then you really need to get half of a 24 or 48 pin printer. Now Star are going to hot up the market by offering a range of 24 pin printers with an output quality that's equal to that of a 48 pin printer. The company is also offering 12 months on-site parts and labour warranty. Until recently, a 48 pin printer could cost you in excess of £2,000. Now Star has brought prices for 48 pin quaity to below £600.

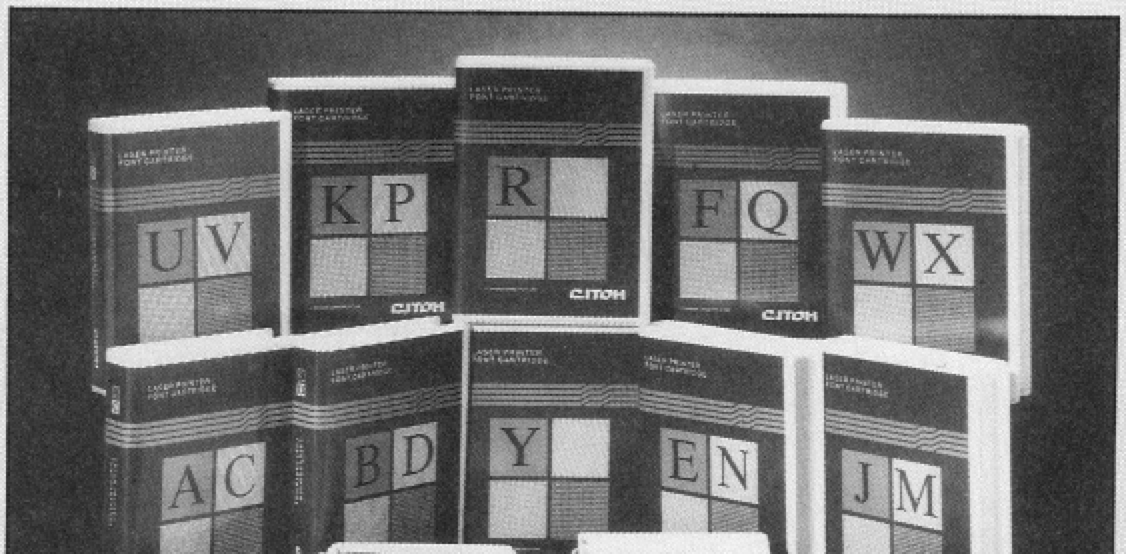
Prices for the new printers are £599 for the 10 inch XB24-10, and £789 for the 15 inch XB24-15. Both these printers have a Super Letter Quality mode offering a character matrix of 48 * 35 dots, the equivalent resolution of many 48 pin printers. A range of 17 fonts are supplied as standard with the printers, and more can be purchased as you require them. A colour option is also available for both printers.

For more information contact Star at Craven House, 40 Uxbridge Road, London W5 2BS. Tel: 01-840 1800.

New Fonts For Lasers

If you can own a HP or IBM laser printer for use with your computer, then the price of new font cartridges has probably put you off buying new fonts for the printer. Now C.Itoh is offering a new range of font cartridges that are compatible with IBM and HP laser printers. They cost around the same price as the "real" cartridges, but offer twice as many options. C.Itoh are also willing to make quotes for people who want their own fonts on cartridge.

The font library consists of 12 cartridges, all ideally suited for use with C.Itoh's C-15 printer and scanner. Once a picture or design has been scanned or merged with text, the whole effect can be maximised by using interesting fonts from C.Itoh's library.



C.Itoh's range of IBM and HP compatible font cartridges - twice as many options

Bloodwych



Bloodwych – adventure in a 3D castle

Fantasy role-playing games are about to become twice as good! That's the claim of Image Works as it adds the final touches to its game *Bloodwych*, which allows two players to control two parties while exploring the same 3D castle. The castle in question is crawling with monsters to slay, tunnels to explore and mysteries to solve, and also hides four crystals that together offer the choice to banish evil or make it prevail forever.

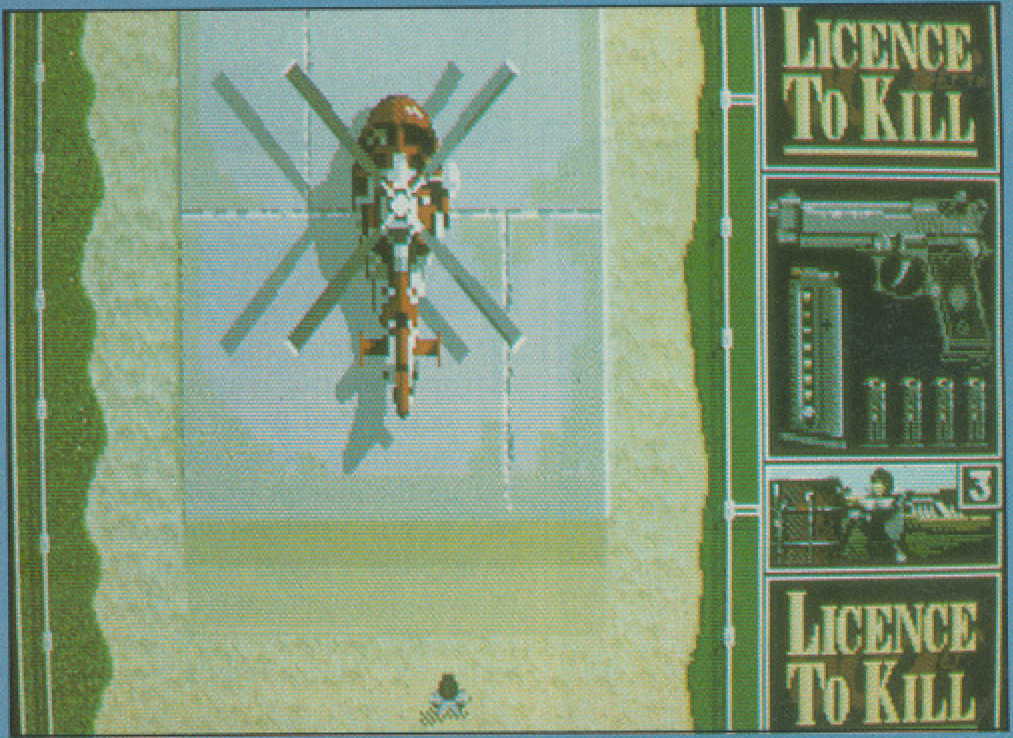
A horizontal split screen will keep the two parties apart and allow them to explore the castle at their own speed, but they are sure to meet somewhere in the darkness.

Adventurers will be able to explore the world of *Bloodwych* on the Amiga from the end of July, and then later on C64 and PC computers.

Bond is Back

As you probably already know, James Bond is back in a hard-hitting new film, and the game of this film will shortly be published by Domark. *Licence to Kill*, that is the film, the C64 game, the Amiga game and the PC game, will all be released simultaneously in June. The new Bond plot centres around 007's personal vendetta against the drug baron Sanchez.

According to Domark, you will mirror Timoty Dalton's actions as Bond in an exhilarating helicopter chase, a death-defying scene in shark infested waters and a final race against time as Bond chases after the escaping Sanchez in a crop duster. Your mission – to destroy Sanchez and the many tentacles (*excuse me! – Ed*) of his drug dealer network.



Licence To Kill – Will Bond nobble the drug dealer?

Microprose marches on

Now that the ink has dried on the sale of Firebrid and Rainbird, Microprose has announced a whole gamut of games from its newly acquired labels, including the C64 *Star Trek* that's reviewed elsewhere in this issue, and many others, including the following:

Weird Dreams, starts with you lying comatose in a hospital bed while your subconscious wanders into a Dali-esque landscape full of your worst nightmares, a place where toys come to life, hideous creatures mutate in front of you and almost everything is out to get you. Even carnivorous roses snap their jaws at you. C64 and PC versions are due in June, and an Amiga version will follow.

Verminator is set inside an ancient oak tree consisting of over 250 locations, each swarming with vermin. Your job is to zap, clonk and knock out every wriggler and flyer you can find. Each kill is worth money that can be saved in the tree's bank (there's a branch near you), gambled in the casino or borrowed from the mob and used to buy equipment that you'll need to reach the higher parts of the tree.

Oriental Games offers four styles of fighting for the price of one. Twenty four computer opponents await your challenge in four mini-tournaments for the championships of Kung Fu, Hollywood Rules, Sumo Wrestling and Kendo. C64, Amiga and PC versions are due in August.

Finally, the world of the cartoon hero comes you your screen, as *Rick Dangerous* battles his way through four levels that span they mysterious Aztec temples and eerie Egyptian tombs. These levels combine to create a game containing 85 screens to challenge the C64 gamer (Amiga owners get 50 more), each filled with traps to avoid, puzzles to solve and guards to destroy, as platform games make a comeback.

The acquisition of these titles means that Microprose can now draw on games from Firebird, Rainbird, Origins and Cosmi to support its own range of simulations.

Psygnosis on PC

Psygnosis, already recognised as one of the major producers of quality Amiga games, is set to launch some of its titles onto the PC market. *Beal, Captain Fizz Meets The Blastertrons, Menace* and *Ballistix*, are all to appear in PC format. C64 users haven't been forgotten either, and *Baal, Ballistix* and *Captain Fizz* will make an appearance in this format.

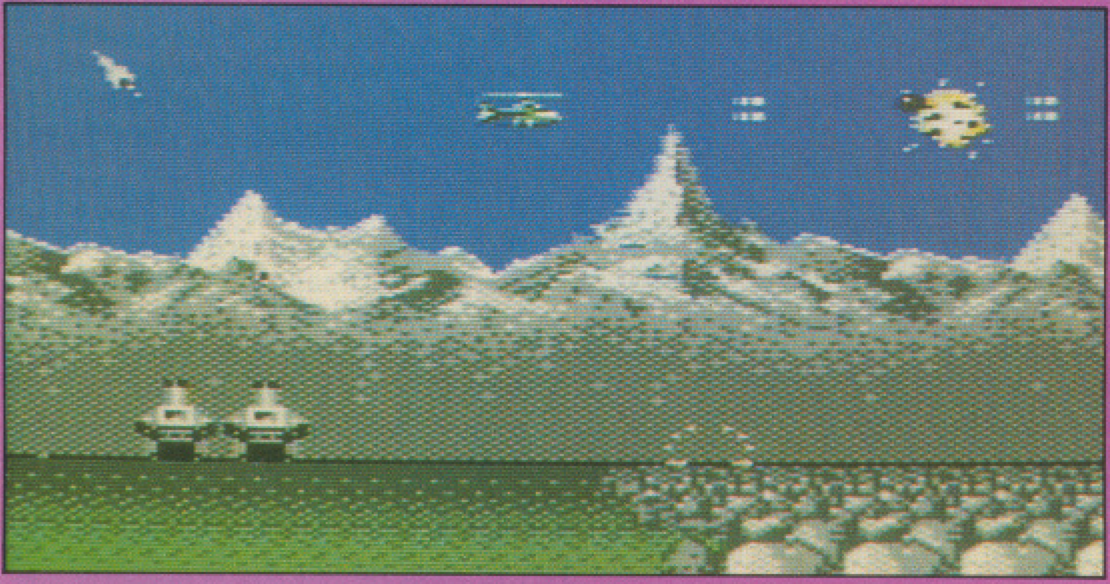
Ocean takes the budget plunge

After months of speculation, Ocean has finally taken the plunge and launched its own budget label. The first six titles from the Hit Squad will include Daley Thompson's *Decathlon, Rambo, Yie Ar Kung Fu, Miami Vice, Green Beret* and *Enduro Racer*. Each game will cost the now standard budget price of £2.99, and is sure to send a tidal wave through the ranks of budgeteers and their armies of ninjas and simulators.

16 bit budgeteers

Kixx, the US Gold budget spin-off, has launched a new 16 bit budget label called Klassix, that aims to star releasing Amiga and PC games for only £9.99. The first batch of three will include the former Christmas number one. *Outrun* (Amiga). Exocet's superb shoot-em-up *Foundation's Waste* (Amiga), and the ultimate in arcade golf games, *World Leader Board* (Amiga, PC).

Kixx is set to continue its C64 range with £2.99 cassette rereleases of *Gauntlet II, Masters of the Universe, Jack the Nipper, Mission Elevator* and *Cybernoid*, as well as £4.99 disk rereleases of *Gauntlet, Super Cycle, 720, Road Runner* and *World Games.*



The excitement of Silkworm – but has the arcade smash survived conversion

Silkworm

Silkworm is the latest arcade smash to be converted for home computers, and features simultaneious two-player action, as you take control of a helicopter and an armoured jeep in a battle against level after level of tanks, helicopters and jets. *Silkworm* is available for C64 (£9.99) and Amiga (£19.99) computers, and marks the return of the Virgin Games label. From now on Virgin Games will be the arcade and sports label for the Virgin/Mastertronic giant that also includes Melbourne House (fantasy and role-playing), Mastertronic (budget) and Leisure Genius (computer versions of classic board games).

Jaws bytes back

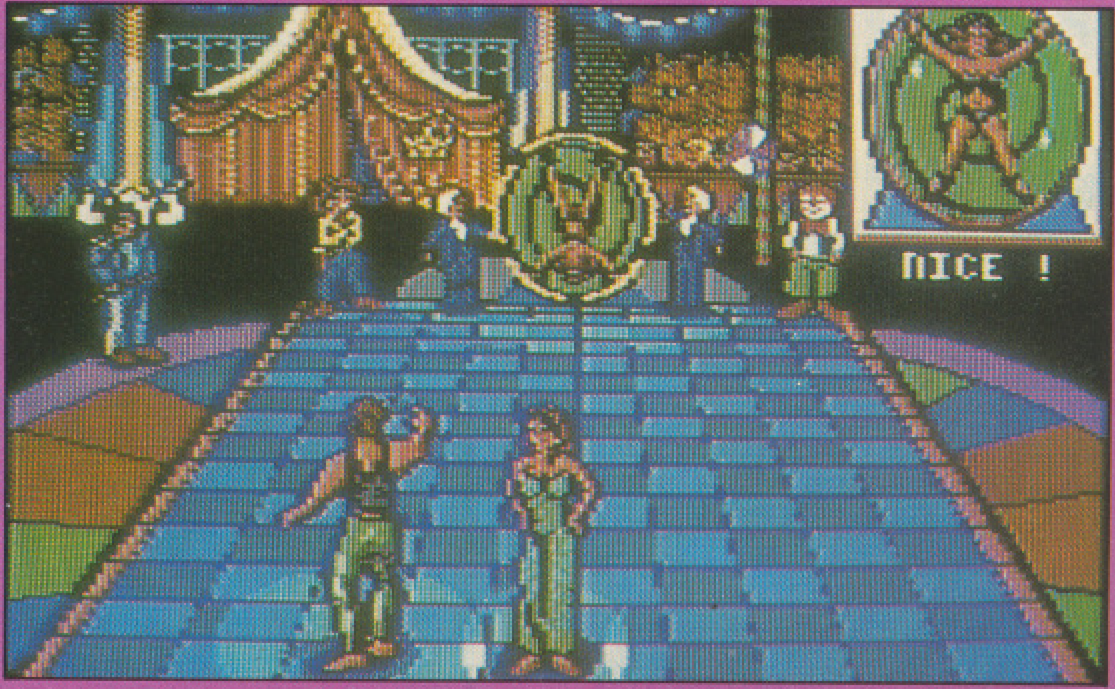
Jaws, the cult movie of the 70's, is at last to feature in a game that looks set to be the debut for a new software house. Screen 7 hopes to recreate the tension and atmosphere as Chief Brody, Hooper (shark expert) and Quint (shark fisherman) set out on the trail of a Great White Shark that's terrorising the inhabitants of Amity island.



Is that a shark or what?

Circus Attractions

All the fun of the circus is set to appear on a C64, Amiga or PC computer near you, thanks to German software house Rainbow Arts. To be released through its Golden Goblins label, *Circus Attractions* features five events that can be played by one or two players. These events include trampolining, juggling, tightrope-walking, knife-throwing and the curious clown jumping, where you're expected to jump between two seesaws as spectacularly as possible. All this action will be presented in what is described as "3D fun, graphics and film comparable animation", backed up with circus-style music.

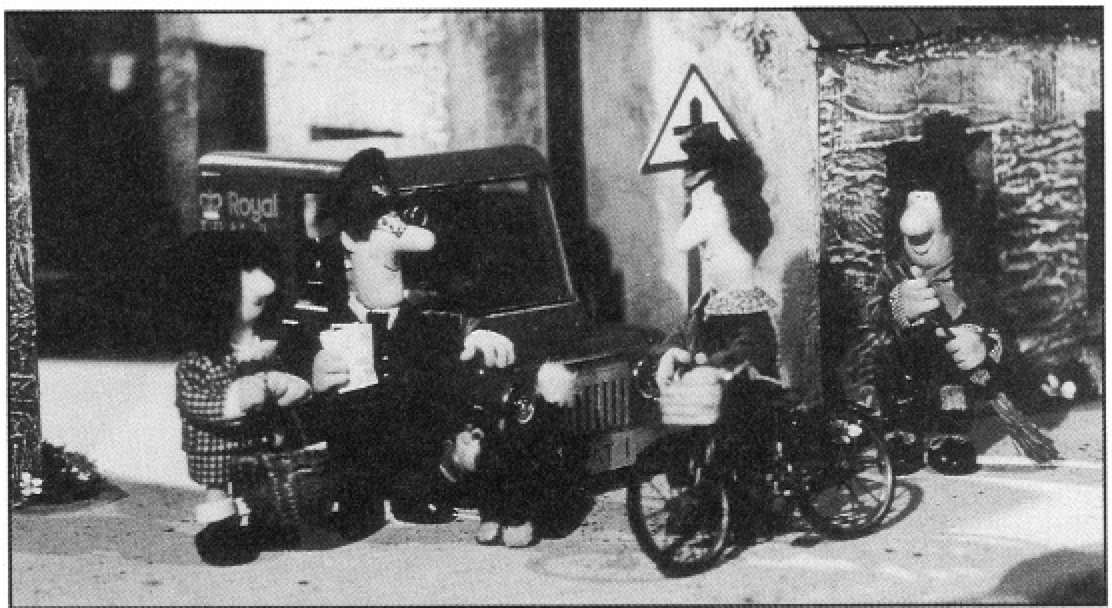


A new day in the Big Top, another new knife-throwing assistant...

Ultima Trilogy

Here's an offer you can't refuse. Not, that is, if you're a role-playing fanatic, as Origin has at last bundled together the first three *Ultimas* to form the *Ultima Trilogy*. This not only represents excellent value for money, but it's also the UK debut of *Ultima II. (Ultima III* was the first *Ultima* launched in the UK followed by *IV, I* and then *V*).

C64 disk (£24.95) and PC (£29.95) owners can battle with the Traid of Evil in three magical games. In *Ultima I – The First Age of Darkness* you must battle with hordes of nightmarish creatures from Mondain the Wizard's Lair. *In Ultima – The revenge of Darkness*, the land is threatened by Minax, Mondain's forgotten apprentice, who has torn rifts in time in her attempt to seek and avenge her father's slayer. These rifts give access through which a brave adventurer may bring about her down. Finally, in *Exodus – Ultima III* a party of adventurers must act quickly, for Sosaria is threatened as the great earth Serpent awakens from a slumber of ages, and fragements of a manuscript hint at an alliance between Mondain and Minax.



Megastar Postman Pat will be appearing shortly on your computer

Postman Pat

Anyone with a younger brother or sister or young son or daughter will know all about Postman Pat and his black-and-white cat. Well, he and the moggy Jess are about to star on the C64 and Amiga screen courtesy of Alternative Software. For only £1.99, C64 owners can deliver the mail in Greendale, meet people like Ted Glen, Peter Fogg, Miss Hubbard and Dr Gilbertson, and also attempt other tasks such as rounding up sheep (*Clive Grace will definitely go for this one! – Ed*).

The Amiga version will follow later (price to be announced), and will also feature Postman Pat Ludo, Snakes and Ladders and Snap. According to Alternative the game is "maddeningly addictive", as is the tune. All together now, "Postman Pat, Postman Pat, Postman Pat and his black-and-white cat..."

Extending Basic

Make your life easier by adding a trace routine to Commodore Basic.

By Burghard-Henry Lehmann

The initial work of writing a program is not all that hard. If you've got a fair grasp of the language you're using and know your computer quite well, you'll get something written pretty quickly. But then comes the laborious bit: testing and debugging!

This gets more difficult and more confusing the larger and more complex your program becomes. A computer program can easily develop into a gigantic jigsaw puzzle, and every bit has to fit exactly! A computer is a machine, and the machine has no mercy. You either get it right or you don't. If you don't, the machine will repeat the same kind of nonsense ten times, a hundred times, a thousand times. Computer novices (and often programmers, who really should know better) when confronted with a bug, operate the program ten or 20 times in the vain hope that the computer will do it right eventually. But it never does!

A far more sensible approach is to get some good debugging tools. Commodore Basic has no debugging tools at all. So, in the next few articles in this series on extending the Basic of your C64, I'd like to develop a few of those. Once you know how Basic works in Rom, this becomes easier than you might think! Let's start with a trace routine.

All About Tracing

Tracing gets activated once the program starts to execute, that is, after you've given the Rom command. Before each line (or part of a line, if it is a multistatement line) is interpreted and executed, the computer is stopped and the trace routine, which we will develop, prints the number of the line to be executed and its contents at the top of the screen.

To continue execution, just press any key – this results in the line being executed as usual. Then the computer jumps to the next line, prints that one at the top of the screen, stops again and so on. The usefulness of all this is that you can see exactly where in the program the computer is at all times. This is also called single-stepping a program, because that's exactly what the computer is doing. Normally, things happen so quickly that you barely have time to consider what exactly is going on. Tracing or single-stepping gives you the chance to take it step by step and think things through.

Stopping the C64 in Its Track

We intercept the normal program flow right at the beginning of our extended Basic routine before we look for an extended Basic command (lines 490-510). At this point, the accumulator contains the first letter of the extended Basic command or the token code, if it is an ordinary Basic command. Since we need that later on, we save it on the machine stack. Then we call the trace routine itself (lines 1360-320).

First we test the system variable $9D, which tells us whether we're in program execution mode or direct mode. If the computer is in the program mode, $9D contains zero, otherwise it contains 128. We want to know this, because if we've just given a Basic command direct, we obviously don't need the trace facility to be activated. Therefore, we exit straight away from the routine (line 1430), recover the former contents of the accumulator from the machine stack (line 510), and continue as usual.

Plotting

Next, we need to save the current print position, because if the program outputs text or graphic characters to the screen, we want them afterwards to be output to the proper print position.

By the way, I've only bothered to save the current print position and recover it later on. To make this program more functional, I advise you to save the screen colours too and print

the tracing line in a way which makes it stand out nicely. As always, I leave these finer points to you!

To save the current print position and initiate our own, we use a Rom routine, called "Plot". If Plot is entered with the carry flag set, the current print position contained in the system variables $D6 and $D3 is put into X (across) and Y (down).

If Plot is entered with the carry flag clear, the value contained in X is initiated as the new print column and the value contained in Y is initiated as the new print line. This knowledge should make it easy for everybody to develop an "AT" function, something sadly lacking in Commodore Basic! In lines 1470-1500, we use plot to save the current print position in 251/252, and in lines 1540-1570 we initiate the top line of the screen as the new current print position.

Next, we print an 80 characters-long empty string to clear the top two lines of the screen (lines 1610-1630). For this we use a Rom-routine which prints any string, as long as it doesn't exceed 256 characters and is terminated with a zero. To point the computer to the string we want to print, we put the low byte of the start address of the string in the accumulator and the high byte into Y. I've decided to clear two lines, to accommodate a Basic line of any length.

Then we reset the current print position back to the beginning of the top line (lines 1670-1700). Printing the line number, which is the first thing we want to do, is also very easy: the number of the line the computer is interpreting at present is contained in the system variable $39/3A. To print that number, we use a Rom-routine which prints any number if the accumulator contains the low byte of the number, and X its high byte.

Next, we print a separating space, using the easiest-to-use of all Rom-routines. With this one, you simply load the character to be printed into the accumulator and call the routine. You don't have to save any registers, because this routine saves everything, including the accumulator which contains the character to be printed, before it does its work and recovers everything again, (with most other Rom-routines you have to take care of this yourself!).

By the way, both the above routine and the Printstring routine also execute so-called "non-printable" characters, such as carriage returns, backspace, cursor movements and so on. Just put the appropriate ASCII code (not the Commodore code!) into the accumulator or the string you want printed, and it does it.

Basic Token Codes

In past articles, I have already mentioned that Commodore compresses (or "tokenized", as the computer jargon goes) Basic keywords. One advantage of this is that it makes Basic textfiles more compact and thus saves memory. It also tells the computer very easily when it has to deal with a Basic keyword and when it doesn't. A token code is always larger than 128, while an ordinary letter or number is smaller than 128. To put it differently, with a token code, bit 7 of the eight bits of a byte (counted from 0 to 7) is set, while with an ordinary ASCII code it's clear.

Of course, this means that a Basic keyword has to be tokenized before the line is entered into the textfile, and every time the line is reprinted on the screen each token has to be expanded into the keyword it stands for. When interpreting the program, the computer just uses the token code, and never expands it, because computers, unlike human beings, are far happier just to deal with numbers, rather than English words.

To expand token codes, there is a list in Rom of all Basic keywords. This list starts at location $AO9E. Since all Basic keywords are of various lengths, the programmers of the Commodore Rom separated each keyword from the next one by adding, once again, 128 to the last letter. Therefore, to print the last letter correctly, one has to subtract 128 from it.

Instead of this method, the Commodore whizkids could have used another method to build this table – they could have padded each keyword that's shorter than a certain length with spaces or zeros. This would have made it much easier to jump from entry to entry. But it would also have used up much more memory. That's why they went for the former solution.

To expand and print token codes we use a subroutine (lines 2660-3200), because there may be several token codes in one line or line segment. This is because Basic tokenizes not just commands, but also functions. It also tokenizes arithmetical operators, like "+" and "–". The reason for this is again to make them stand out clearly from other characters in the line.

Expanding and printing a token

When we enter our subroutine, the token code is the accumulator. First, we subtract 128 from the code to get the actual number of the keyword. This we load into X, which will serve us as the counter (lines 2660-2680).
Next, we load the base address of the keyword table ($AO9E) into a zero page address so we can use indirect-Y later on. I use zero page $61/62, which is the first location of the floating point accumulator. It is completely safe to do this here, since we won't use the floating point accumulator (lines 2720-2750).

Now we enter the main loop, which starts by testing X. If it contains zero, we have found the keyword we were looking for. This is because we use X, which contains the number of the keyword, to count backwards. With every pass through our main loop, we decrement X by one. (lines 2790-2810)

If we haven't found our keyword yet, we zero Y and enter TOKENLP1 (lines 2850-2900). This loop tests each character of the next keyword to find the last character, which has 128 added to it. If it has found that character, the carry flag will be set. In the final part of the main loop (lines 2940-3010), we update the base address contained in $61/62 so that it points at the beginning of the next keyword. This is done by adding the contents of Y (the index) to it.

Once we have found the right keyword, it's printed onto the screen, again each characer being tested to find the last character (lines 3050-3110). When this has been found, 128 is subtracted from its value and it too is printed (lines 3160-3180).

Tying it all up

The rest of our trace routine is pretty simple.
Lines 2410-2440 reinstate the former current print position.
Finally, a Rom-routine called "Getin" is used to wait for any key being pressed (lines 2480-2490). If no key has been pressed, the zero flag is set. Otherwise zero will be clear, because Getin returns the ASCII code of the

key which has been pressed in the accumulator.

As always in this series, I haven't done things as comprehensively as they could have been – I justify this by saying that I want to encourage you to find your own solutions, but people who know me better will say that it's because I'm bone idle... I haven't added a new basic command which switches trace on and off. I'm sure you may want to introduce these commands, since they allow you to trace through certain parts of a program and let other parts you are less interested in run at full speed.

Next time we'll develop some more debugging and toolkit routines.

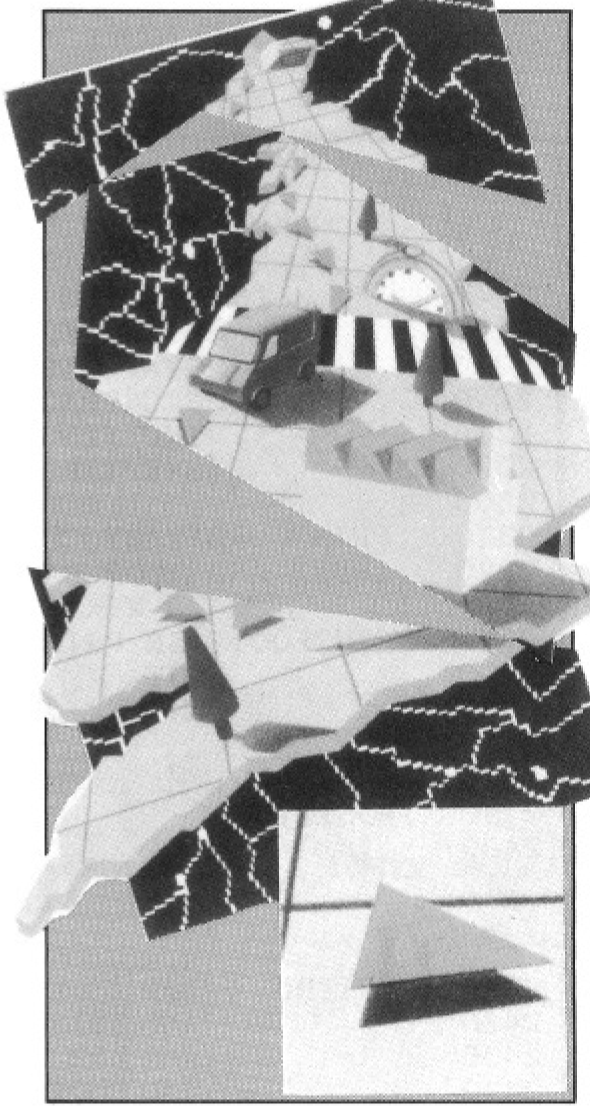
```
10               ORG 49152
20               ENT
30   ;
40   CHARGET     EQU $0073
50   EXECVECT    EQU $0308
60   PRINT       EQU $E716
70   PRINTNO     EQU $BDCD
80   PRINTSTR    EQU $AB1E
90   PLOT        EQU $FFF0
100  ;
110  SYMBOLTBL EQU 50000
120  ;
130  ;
140  ;
150  ;TURN EXTENDED BASIC ON
160  ;BY CHANGING VECTOR AT $0308
170  ;
180  EXTBASON    LDA #<PRGSTART
190              STA <EXECVECT
200              LDA #>PRGSTART
210              STA >EXECVECT
220  ;
230              RTS
240  ;
250  ;
260  ;
270  ;TURN EXTENDED BASIC OFF
280  ;BY CHANGING VECTOR AT $0308
290  ;BACK TO NORMAL ($A7E4)
300  ;
310  EXTBASOFF LDA #<$A7E4
320              STA <EXECVECT
330              LDA #>$A7E4
340              STA >EXECVECT
350  ;
360              RTS
370  ;
380  ;
390  ;
400  ;*** MAIN PROGRAM ENTRY ***
410  ;
420  ;LOOK FOR EXTENDED BASIC COMMANDS
430  ;
440  PRGSTART    JSR CHARGET
450              JSR EXECSTM
460              JMP $A7AE
470  ;
480  ;
490  EXECSTM     PHA
500              JSR TRACE
510              PLA
520  ;
530              CMP 'O
540              BNE NEXT
550              JMP OFF.RT
560  ;
570  NEXT        CMP 'C
580              BNE NORMAL
590              JSR CHARGET
600              CMP 'O
610              BNE NORMAL
620              JSR CHARGET
630              CMP 'L
640              BNE NORMAL
650              JSR CHARGET
660              CMP #$B0      ;'OR' TOKEN
670              BEQ COLOR.RT
680  ;
690  ;
700  ;DO NORMAL ROM-ROUTINE
710  ;
720  NORMAL      JMP $A7ED
730  ;
740  ;
750  ;EXECUTE 'COLOR' COMMAND
760  ;
770  ;GET INK PARAMETER
780  ;
790  COLOR.RT    JSR CHARGET
800              JSR $AD8A
810              JSR $B7F7
820  ;
830  ;CHANGE INK COLOUR
840  ;
850              STY 646
860  ;
870  ;GET PAPER PARAMETER
880  ;
890              JSR CHARGET
900              JSR $AD8A
910              JSR $B7F7
920  ;
930  ;CHANGE PAPER COLOUR
940  ;
950              STY 53281
960  ;
970  ;GET BORDER PARAMETER
980  ;
990              JSR CHARGET
1000             JSR $AD8A
1010             JSR $B7F7
1020 ;
1030 ;CHANGE BORDER COLOUR
1040 ;
```

```
1050              STY 53280
1060    ;
1070    ;JUMP TO REST OF ROM-ROUTINE
1080    ;
1090              RTS
1100    ;
1110    ;
1120    ;
1130    ;TEST FOR REST OF 'OFF'
1140    ;
1150    OFF.RT    JSR CHARGET
1160              CMP 'F
1170              BEQ OFF.RT1
1180              JMP NORMAL1
1190    OFF.RT1   JSR CHARGET
1200              CMP 'F
1210              BEQ OFF.RT2
1220              JMP NORMAL1
1230    ;
1240    ;EXECUTE 'OFF' COMMAND
1250    ;
1260    OFF.RT2   JSR EXTBASOFF
1270    ;
1280    ;GET NEXT CHARACTER AND
1290    ;JUMP TO REST OF ROM-ROUTINE
1300    ;
1310              JSR CHARGET
1320              RTS
1330    ;
1340    ;
1350    ;
1360    ;TRACE ROUTINE:
1370    ;
1380    ;IF DIRECT MODE, EXIT AT ONCE.
1390    ;
1400    TRACE     LDA $9D
1410              CMP #$80
1420              BNE TRACE1
1430              RTS
1440    ;
1450    ;SAVE CURRENT PRINT POSITION.
1460    ;
1470    TRACE1    SEC
1480              JSR PLOT
1490              STX 251
1500              STY 252
1510    ;
1520    ;PLOT TOP LINE PRINT POSITION.
1530    ;
1540              CLC
1550              LDX #0
1560              LDY #0
1570              JSR PLOT
1580    ;
1590    ;CLEAR TOP TWO LINES.
1600    ;
1610              LDA #<EMPTYLINE
1620              LDY #>EMPTYLINE
1630              JSR PRINTSTR
1640    ;
1650    ;PLOT TOP LINE PRINT POSITION.
1660    ;
1670              CLC
1680              LDX #0
1690              LDY #0
1700              JSR PLOT
1710    ;
1720    ;PRINT LINE NUMBER.
1730    ;
1740              LDA $3A
1750              LDX $39
1760              JSR PRINTNO
1770    ;
1780    ;PRINT ONE SPACE.
1790    ;
1800              LDA #32
1810              JSR PRINT
1820    ;
1830    ;GET ADDRESS OF BASIC TOKEN AND
1840    ;PRINT IT.
1850    ;
1860              LDY #0
1870              LDA ($7A),Y
1880    ;
1890              JSR TOKENSR
1900    ;
1910    ;PUT CHARGET ADDRESS INTO 253/254
1920    ;AND INCREMENT BY ONE.
1930    ;
1940              LDA <$7A
1950              STA <253
1960              LDA >$7A
1970              STA >253
1980    ;
1990              INC <253
2000              BNE TRACE2
2010              INC >253
2020    ;
2030    ;PRINT REST OF LINE.
2040    ;
2050    TRACE2    LDY #0
2060    PRINTLOOP LDA (253),Y
2070              BEQ LINEEND
2080              CMP ':
2090              BEQ LINEEND
2100              CMP #128
2110              BCC PRINTLOO1
2120    ;
2130    ;IF TOKEN, SAVE Y AND PRINT TOKEN
2140    ;
2150              INY
2160              STY $63
2170    ;
2180              JSR TOKENSR
2190    ;
2200    ;POINT AT CHARACTER AFTER TOKEN
2210    ;AND LOOP BACK.
2220    ;
2230              CLC
```

```
2240                LDA $63
2250                ADC <253
2260                STA <253
2270                BCC TRACE2
2280                INC >253
2290                BNE TRACE2
2300  ;
2310  ;PRINT ORDINARY CHARACTER, INCR.
2320  ;INDEX AND LOOP BACK.
2330  ;
2340  PRINTLOO1 JSR PRINT
2350                INY
2360                BNE PRINTLOOP
2370  ;
2380  ;END OF LINE: RE-PLOT OLD PRINT
2390  ;POSITION.
2400  ;
2410  LINEEND   CLC
2420                LDX 251
2430                LDY 252
2440                JSR PLOT
2450  ;
2460  ;WAIT FOR KEYPRESS.
2470  ;
2480  WAIT      JSR $FFE4
2490                BEQ WAIT
2500  ;
2510  ;IF KEY PRESSED, EXIT.
2520  ;
2530                RTS
2540  ;
2550  ;
2560  ;
2570  NORMAL1   JMP $A7ED
2580  ;
2590  ;
2600  ;
2610  ;TOKEN SUBROUTINE:
2620  ;
2630  ;CALCULATE TOKEN NUMBER AND
2640  ;STORE IT IN X.
2650  ;
2660  TOKENSR   SEC
2670                SBC #128
2680                TAX
2690  ;
2700  ;INITIATE BEG. OF TOKEN TABLE.
2710  ;
2720                LDA #<$A09E
2730                STA <$61
2740                LDA #>$A09E
2750                STA >$61
2760  ;
2770  ;IF X=0, TOKEN FOUND.
2780  ;
2790  TOKENLOOP CPX #0
2800                BEQ PRINTOKEN
2810                DEX
2820  ;
2830  ;FIND END OF TOKEN.
2840  ;
2850                LDY #0
2860  TOKENLP1  LDA ($61),Y
2870                CMP #128
2880                BCS ENDTOKEN
2890                INY
2900                BNE TOKENLP1
2910  ;
2920  ;POINT TO BEG. OF NEXT TOKEN.
2930  ;
2940  ENDTOKEN  CLC
2950                INY
2960                TYA
2970                ADC <$61
2980                STA <$61
2990                BCC TOKENLOOP
3000                INC >$61
3010                JMP TOKENLOOP
3020  ;
3030  ;PRINT BASIC WORD.
3040  ;
3050  PRINTOKEN LDY #0
3060  PRINTOK1  LDA ($61),Y
3070                CMP #128
3080                BCS TOKENEND
3090                JSR PRINT
3100                INY
3110                BNE PRINTOK1
3120  ;
3130  ;PRINT LAST CHAR. OF BASIC WORD,
3140  ;AFTER HAVING SUBTRACTED 128.
3150  ;
3160  TOKENEND  SEC
3170                SBC #128
3180                JSR PRINT
3190  ;
3200                RTS
3210  ;
3220  ;
3230  ;
3240  EMPTYLINE BYT "                                        "
3250                BYT "                                        "
3260                BYT "                                        "
3270                BYT "                                        "
3280                BYT "                                   ",0
```

Autoroute

Tony Heatherington discovers a unique new pathfinding system that could make life easier for tourists and travelling salesmen everywhere



Maps are supposed to be easy to read, making it easier for you to plan your journey. But how often has the place you've been aiming for been obscured by a fold or a staple? How often have you tried to find your way around "proposed motorway routes", or stared at your map trying to find a path around some particularly heavy roadworks or locate a place to stop and eat on the way? Wouldn't it be good if you could use a computer to plan your routes by simply typing in where you wanted to go? Such a system is no longer confined to the realms of science fiction – it already exists.

Autoroute from Next Base is available in several forms, ranging from the basic *Autoroute* up to the new *Autoroute Plus* and its add-on modules. All you need to run it is a PC with either twin floppies or a hard disk drive, as a detailed map of Britain eats up memory more than anything else.

Using *Autoroute* couldn't be easier – all you have to do is type in your starting location and your destination. If you misspell either place, or if *Autoroute* isn't sure where you mean, it displays a menu of options to choose from. That sounds easy enough, but the real power comes when you can add up to 20 stops on the route, establish the time you'll be travelling, specify road types you prefer or would rather avoid, and rate the speeds you travel at somewhere between 2Cv and GTi.

Autoroute then uses this information to calculate a series of routes, and rates them as quickest or shortest using you to choose the best option (the first is usually the quickest and the shortest). You can then display a map of the route and a table of directions.

The map varies in quality depending on your graphics card, is functional in CGA and impressive in EGA. Through a series of key presses and a mouse pointer you can identify unmarked roads, zoom in and out of the map and increase or decrease its detail. In fact you can flood the display with so many place names that it's impossible to see the route.

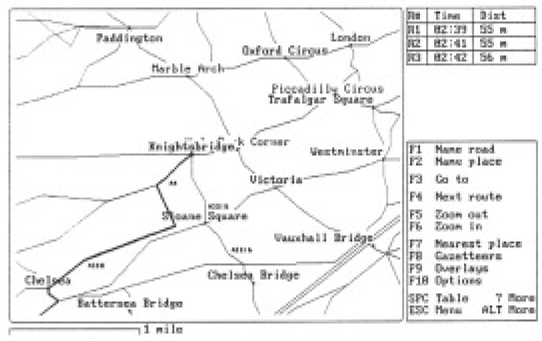
Pressing the space bar takes you to the table or list of directions that you can print out and bring with you in the car. These directions tell you which turnings to take, the distance from the last junction, the direction you're heading in and even the time at which you should reach it.

Although this is ideal for a navigator to read, it would be very dangerous for a driver to refer to such crammed print while in motion, so it's a shame there isn't an option to print the directions in easy to read double-sized print that you could clip onto the dashboard.

Unlike a static map, *Autoroute* can respond to exceptional circumstances. For example, in bad weather you could tell it to avoid B roads where possible, and on Bank Holidays to steer clear of packed motorways. Similarly, you could compensate for major roadworks and other hazards that might slow your journey down.

As if that isn't enough, Next Base has just released an updated version of *Autoroute*, with added features and modules, that's logically called *Autoroute Plus*. *Auto Plus* retains the same easy route-planning structure, but adds new features and options. For example, by clicking on the map you can dodge a specific stretch of road to avoid road works, a snarl up or an accident, or find the nearest pub, hotel or branch of your business.

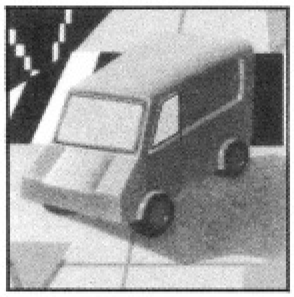
You can also specify the time you need to reach your destination at, and *Autoroute Plus* will plan your journey accordingly. Say, for example, you wanted to travel from Newport on the isle of Wight to Bristol, and travel via Bournemouth. Southampton and Winchester, but reach Bristol for a meeting at 3PM. *Autoroute Plus* does the rest, caluclating and displaying that you must leave Newport at 11am to arrive at Bournemouth at 12:22,

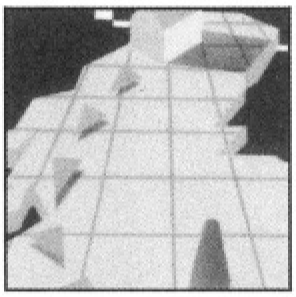


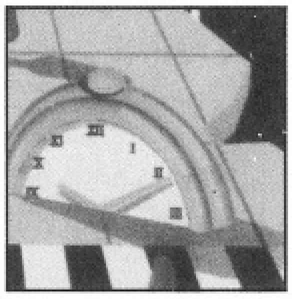
Southampton at 12:57 and Bristol at 3pm. Try working that lot out on a normal road map.

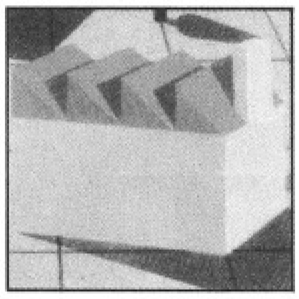
Both the map and the table of directions can be printed out or exported as PCX files and used in DTP packages, so you could tell all your customers where your shop is and include a map and directions of how to get to it in the same document as your latest price list and details of special offers. Similarly, you could use it to find sales or exhibitions, ranging from the PC Show at Earl's Court down to a car boot sale at the local school.

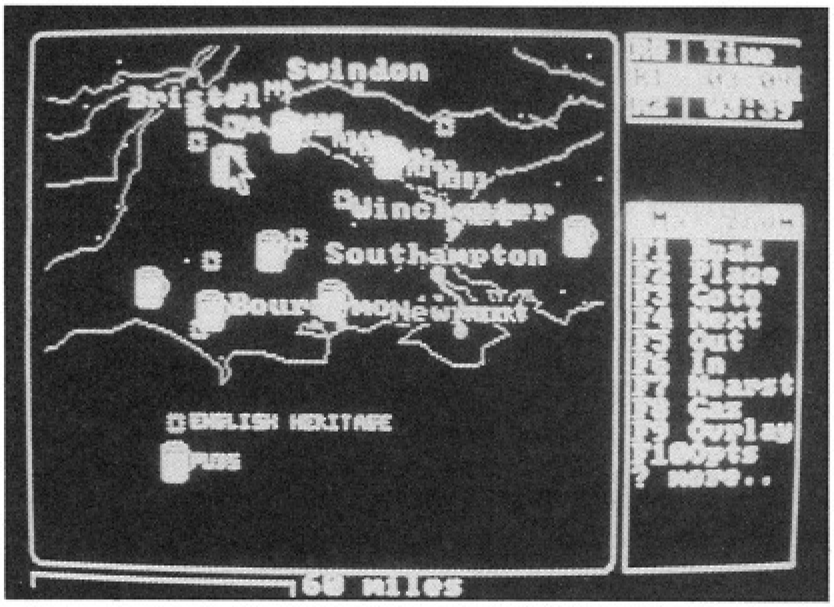
Autoroute Plus is an open-ended system that can be updated and further expanded through a series of modules. These include a Gazetteer Editor, Postcodes and optimisation and costing systems. For the general user the Gazetteer shows most promise, as it allows you to add places of interest

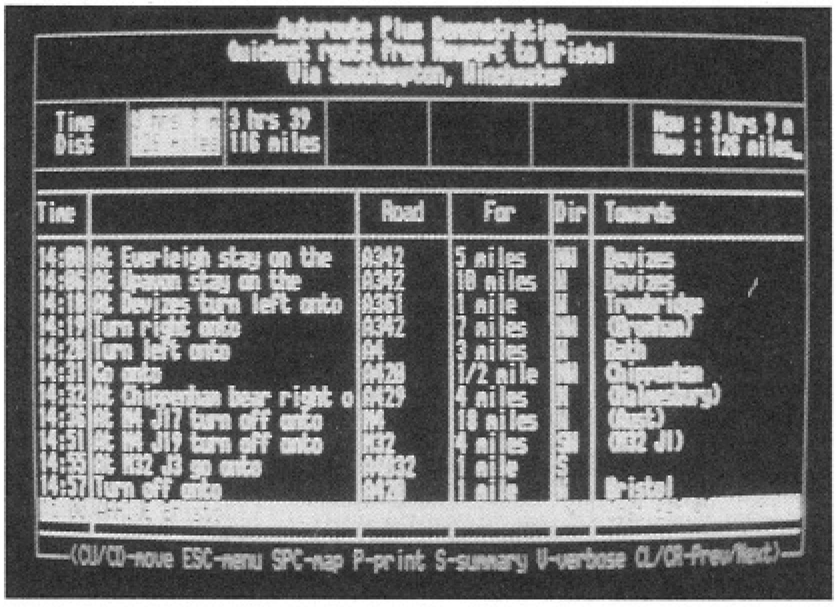


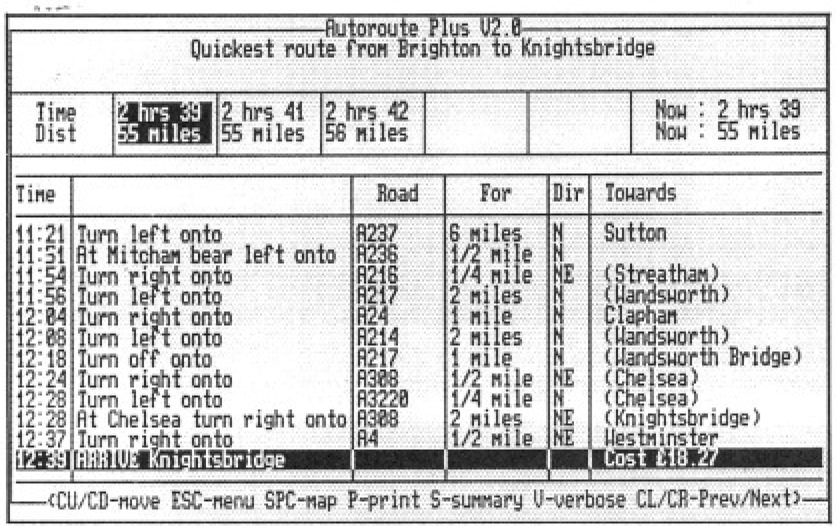










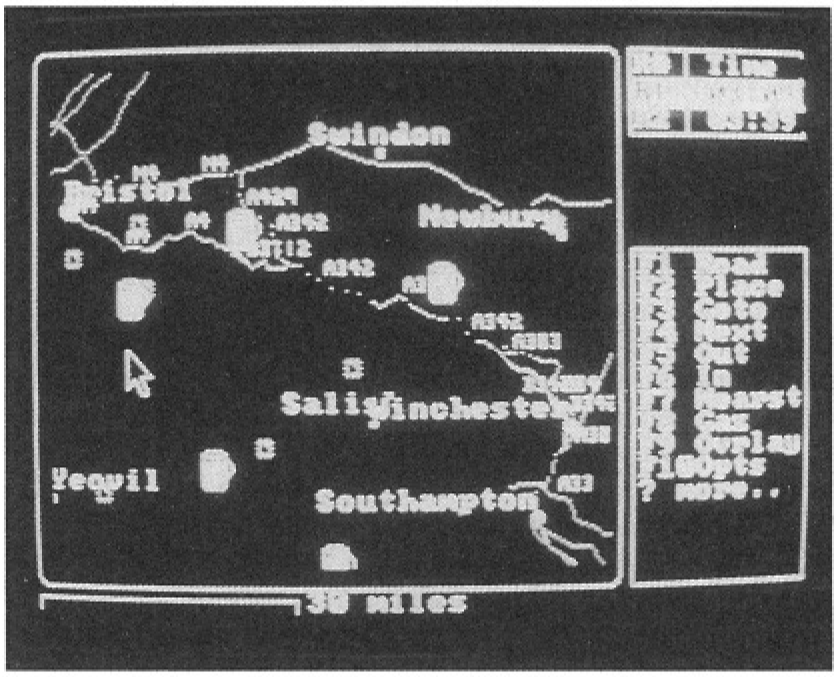


Autoroute Plus V2.0
Quickest route from Brighton to Knightsbridge

Time Dist	2 hrs 39 55 miles	2 hrs 41 55 miles	2 hrs 42 56 miles			Now : 2 hrs 39 Now : 55 miles

Time		Road	For	Dir	Towards
11:21	Turn left onto	A237	6 miles	N	Sutton
11:51	At Mitcham bear left onto	A236	1/2 mile	N	
11:54	Turn right onto	A216	1/4 mile	NE	(Streatham)
11:56	Turn left onto	A217	2 miles	N	(Wandsworth)
12:04	Turn right onto	A24	1 mile	N	Clapham
12:08	Turn left onto	A214	2 miles	N	(Wandsworth)
12:18	Turn off onto	A217	1 mile	N	(Wandsworth Bridge)
12:24	Turn right onto	A308	1/2 mile	NE	(Chelsea)
12:28	Turn left onto	A3220	1/4 mile	N	(Chelsea)
12:28	At Chelsea turn right onto	A308	2 miles	NE	(Knightsbridge)
12:37	Turn right onto	A4	1/2 mile	NE	Westminster
12:39	ARRIVE Knightsbridge				Cost £18.27

<CU/CD-move ESC-menu SPC-map P-print S-summary V-verbose CL/CR-Prev/Next>



to the *Autoroute* maps. The examples supplied are pubs and hotels, but you could add anything from distributors to branches and software houses to customers.

For each entry, you can compile a small text entry that appears when the location is selected by mouse. For example, clicking on the software house in Staines would reveal something like... Next Base, authors of *Autoroute* and *Autoroute Plus* and its expansion modules. Tel: 0784 460077. Fax 0784 460582.

That's not all – the Gazetteer also allows you to add detailed departure and arrival instructions that are incorporated into the table so that people will always know how to find you.

The postcodes module includes the location of the centre of each of the 8700 post code regions, so that you can pinpoint your customers accurately and efficiently. This efficiency can be enhanced by adding the optimisation and costing module, that will automatically plot the most efficient route between calls and calculate costs based on hourly and fuel-dependent rates.

There are modules on the way to customise *Autoroute Plus* to your specific needs – these will include a symbol editor to add symbols to the maps (for use with the Gazetteer), Isochrone, that plots destinations the same travelling time away from your start position (E.G. what's 45 minutes from Birmingham – ideal for distribution), a restrictions module that plans routes to avoid low bridges, width and weight limits, and an overlay of postcode areas and county boundaries.

Autoroute and *Autoroute Plus* will have many users, ranging from computer users wanting to impress family and friends, to travelling salesmen, clubs organising events and small and not-so-small businesses. It will run on most PCs, but becomes really useful when loaded into a portable that can be used while you're in the car.

The basic *Autoroute* costs £130, which is quite reasonable considering the mass of data it stores and processes. *Autoroute Plus* is aimed more at the professional user, and carries a £299 + VAT price tag, with the modules costing between £149 and £199.

Touchline:
Title: *Autoroute.* ***Supplier:*** *Next Base, Unit 18, Central Trading Estate, Staines, Middlesex, TW18 4XE.* ***Tel:*** *0784 460077.*

PC Corner

Commodore has joined just about every other computer manufacturer and started producing PC Clones. We start a regular column for Commodore PC owners.

If you own a Commodore 64, you may be considering changing your machine for an Amiga, or another 16 bit machine. However, you may still be confused as to what's available, or the Amiga may not be the right machine for you. What are the alternatives? The Atari ST is one, but you could also consider getting a PC compatible. This might seem a strange choice, but settling for what has become the de facto industry standard does in fact make a lot of sense. An explanation of why you should take such a course is necessary, so I will attempt to make clear both the pros and cons in this article.

The IBM Standard

The term *IBM PC compatible* refers to a hardware standard, that is a machine which has certain minimum specifications. These are: an 8088 8/16bit CPU running at a speed of 4.7Mhz, a minimum of 64k RAM, a video card capable of displaying at least 80x25 rows of text, and one double-sided 40 track disk drive with 360k capacity. To be fully compatible with the IBM standard, the ROM chips inside the machine should also be the same as IBM's.

You may wonder how manufacturers have survived the threat of legal action from one of the world's biggest companies. The answer is, only at their indulgence, by not making copies directly and by using a technique called 'reverse engineering'. Even basic items such as printer ports are not part of the spec. However, one of the great advantages of the standard is that a large number of expansion slots come in the basic machine, so you can buy almost anything, and just plug it in. Don't worry if you're not quite sure what all this means – everything should become clear as you read on.

MSDOS

Also called PCDos on true IBM machines, MSDos is the heart of a PC compatible. Without it, you cannot do anything at all.

MSDos is usually loaded from disk, but some portables, for example, have a version in ROM. MSDos is responsible for handling all input and output. This means the screen, printer, keyboard or disk drive. The software that does all this is usually loaded once, at switch on and then resides in the machine until it is switched off. The other part of MSDos is made up of a number of *Transient* commands. Of these, by far the most important is **Command-com,** the so-called command processor. This program is like the Basic interpreter, in that it sits in memory intercepting your keypresses, and turning them into something the computer can understand.

All MSDos commands are given from a prompt which initially consists of just the disk drive Letter, and a > symbol eg;

A: > or C: >

This can be changed to suit your needs to show the date, for example. Here you will do most work, launching programs, formatting disks, and keeping track of Data. Typing the built-in command **DIR** at this prompt will reveal what's on disk. *Fig.1.* shows the directory listings of the MSDos system disk. You will see that files on the disk all have up to eight letters, then a dot (.) followed by an extension of up to three letters. How a file is treated by MSDos depends on these letters. **EXE** or **COM** after the dot means a program that will run from the A: > prompt. **BAT** tells MSDos that the file is a text file which can be treated as a list of commands. One of these **.BAT** files, **Autoexec.bat** is special, because it executes automatically on startup. This means you can set your machine up to suit you, in the knowledge that every time you switch it on, it will be the same.

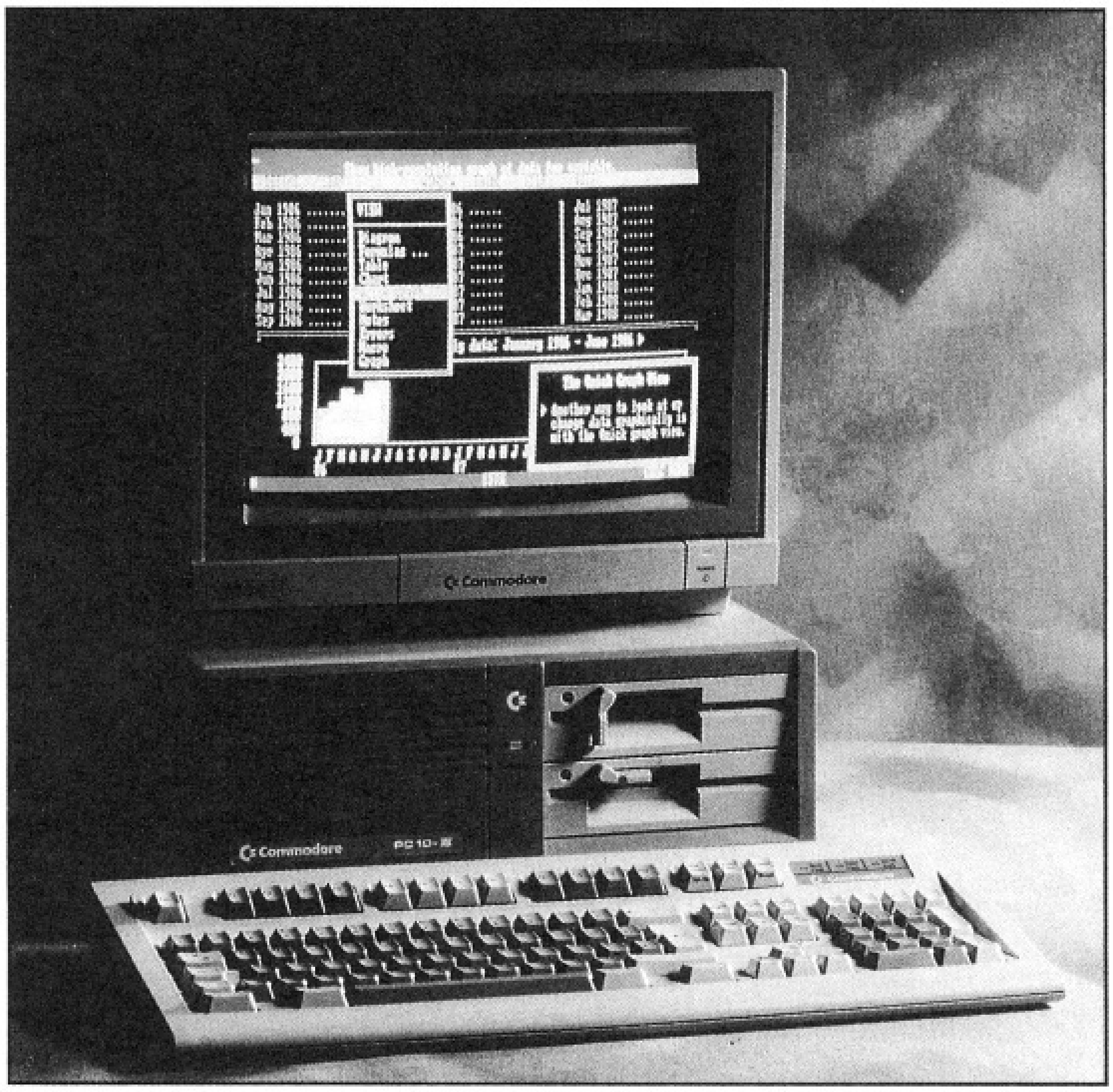
The other important thing to understand about MSDos is that it is a hierarchical filing system, and so within one directory you can have not only files, but more directories, and so on. Organising files this way is essential if you have a hard disk, of which more later.

Programming the PC

The surprising thing about PC compatibles is they almost never have Basic built into them. Instead Basic comes as just another program that can be run. *GWBasic* is generally accepted as the standard here and comes with most machines. An interpreted language, it is generally considered to be well featured and easy for beginners to get to grips with, but it is slow. However, numerous languages are available, and if you wish to program professionally you should consider one of the many compilers available. Assembly language programming is also well catered for, both commercially, and in the public domain. PCs do of course have the *BATCH* language built into them. This is very simple, but experienced batch language programmers can do some surprising things with it.

Software

Upwards of one million packages have been written for the machine, and this number is growing all the time. All the big software houses write for the



PC, and some packages such as *Lotus 123, DBase III*, and *Wordperfect* are standards in their own right. Standards are high, but be warned, unless you have an expensive machine capable of running *Windows,* most software is still text-based. The quality of games software can be variable with only the more recent stuff for EGA graphics adaptors being really exciting. This will improve as more software houses take to the PC. Public domain and shareware is available in large quantities, so even the impecunious will find something to meet their needs.

Graphics

Probably the most perplexing aspect of PCs is that of graphics and display adaptors. Unfortunately there is no single standard, and of those which do exist there are many variations. Also, many programs such as Aldus Pagemaker require a minimum configuration before they will run. In *Pagemaker's* case an *EGA* adaptor and colour monitor are required before it will run.

The problem has arisen because, in the beginning the display standard was 80x25 text in monochrome. Only later did provision for colour and graphics arise. All PC's come with some kind of graphics adaptor, usually on a plug-in card and a monitor, so it is important to get the right display for your needs. The following list should give an idea of what's availble.

The list shows just how desirable some combinations can be.

The drawback however is cost. With a suitable VGA setup costing as much or more than most low cost PC's, many users have to settle for something less. If you can afford it, I would suggest an EGA to be the best overall value for money.

What to look for when buying a PC

Thankfully, modern PC compatibles far exceed the specifications of their ancestors. You should expect memory of 512 or 640K.

A turbo processor running at 8 or 10Mhz is advisable. For those with ample cash, an *AT* class machine which uses the fast 80286 processor could be acquired. Serial and parallel ports should be built in and at least two or three expansion slots should be available.

As Personal Computers are disk based, a minimum of two 360k 5 1/4 inch, or one 720k 3.5 inch drive should be included. Hard disks are cheap in the PC world, and are really worth the money at about 200 pounds for 20Mbytes.

Check also what software is bundled with the machine. An integrated package such as *Ability*, or *Works* could be all you need for the first six months or so.

Conclusion

PC compatibles are a safe option. You won't ever set the world alight with one, but at the same time nobody is going to laugh at you for getting one. A very wide range of price and performance is covered, so the chances are there will definitely be one to match your budget.

Commodore have a range typical of many manufacturers, ranging from budget 8088 machines, to fast 80386's with huge amounts of memory and disk storage.

Get In Touch

PC Corner is designed to be a forum for all users of the Commodore PC range, but it's important to remembe: that without your input, it simply won't work. We want PC Corner to work, but we need your help, so if you have any comments, tips or general PC queries, please get in touch with us at:

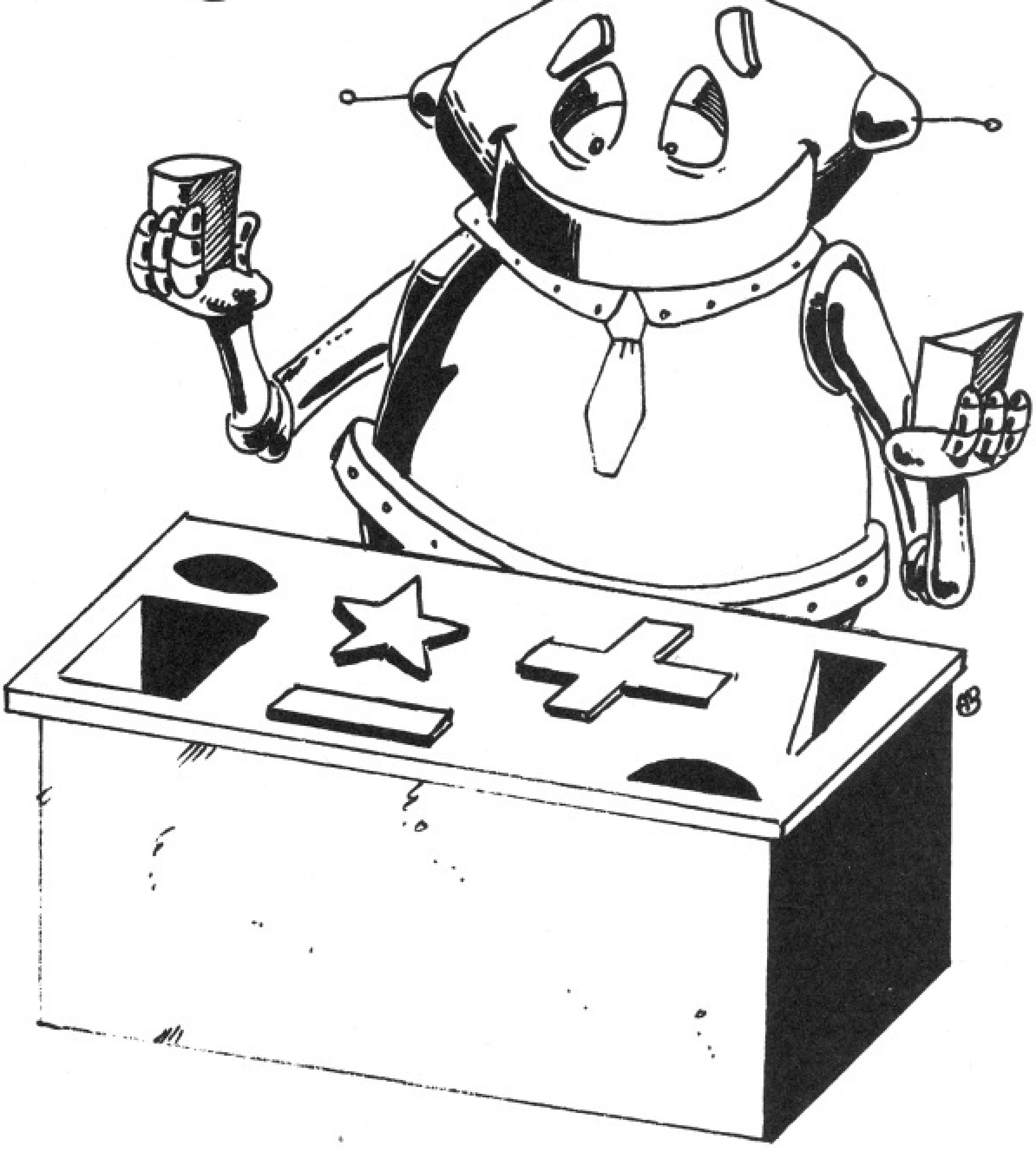
PC Corner
Your Commodore
Argus House
Boundary Way
Hemel Hempstead
Herts
HP2 7ST

Adaptor	Resolution	Use
MDA	80*25(text)	Wordprocessing etc
HGA	as above plus 720x348*2 col	as above plus CAD etc
CGA	320*200*4 & 640*200*2 col	general & games
EGA	up to 640*350*16 col	Hi res graphics/games
VGA	up to 640*480*64 col	DTP etc

Reasoning on the 128

The first part of a series that may help you turn your humble 128 computer into an expert system

By Paul Schofield



I find it surprising that there is still very little AI software available for Commodore Machines. One likely reason for this phenomenon is that many people have been given the impression that AI applications can only be programmed in either PROLOG or LISP. Whilst these languages are certainly very good tools for this type of application, it's quite feasible to employ almost any programming language. A second problem is that AI applications tend to be very data intensive, and the combination of limited memory and slow disks is somewhat restrictive.

Despite this, examples of modern AI techniques are used in many C64 adventure games, some of which feature quite sophisticated natural language processors. In this article we will look at another application area, the Deterministic Expert System. All the programs are written entirely in Commodore Basic 7.0, but allow quite a wide range of simple Expert Systems to be created. Before looking at the first program, however, it's quite natural to look at the features of traditional AI tools and the typical applications of expert systems.

PROLOG

PROLOG is quite unlike Basic and other popular computing languages. Whereas traditional programming languages are concerned with arithmetic calculations, PROLOG is designed to tackle problems by logical deduction. The language contains two basic types of statement. The first type is used to establish relationships between data items, which are then used to create a so called Knowledge Base. The second type of statement is then used to interrogate the Knowledge Base to determine answers to particular problems.

The great power of PROLOG is that it doesn't need to understand the relationships defined. For example, if a restaurant wants to organise its menu to avoid dishes that include out of season fruits and vegetables, they can use statements like:

expensive (strawberries, january).

This condition can then be tested by:

? - expensive (strawberries, january).

to which PROLOG will answer YES. Not very exciting, but if strawberries is replaced by a variable, then it's possible to look at all things which are expensive in January.

LISP

LISP stands for LISt Processor, and can perform similar types of operation,

but in this case all the conditions must be presented in the form of a list. The language provides operations for moving through the list structure, and taking different paths according to the value of individual elements in the list. For a typical AI application a LISP list looks rather bewildering, with many layers of nested brackets used to define the hierarchy of sub-lists. However, this approach is conceptually simpler, as each list element at the lowest level is an English language statement.

These are the questions the user is asked and the user's responses (TRUE or FALSE) determine the next list element presented. ICPUG members, who are interested in LISP should request disk CL15 from the C128 library, which contains a C/PM Mode Lisp Interpreter. Be warned though, you won't get very far without a text book. Alternatively, you can find quite similar facilities within the C64 Logo, together with a very incomplete example of Knowledge Base on the Utility Disk.

What's an Expert System?

While the two favourite languages of the AI community can provide many useful clues, it's also useful to look at the basic concept of an Expert System. As the name suggests, the best way to start is to find someone with a lot of experience of tackling a complex problem. You then spend a good deal of time asking them to run through how they approach the problem step by step.

This will highlight a sequence of questions they ask or tests they perform, and how they proceed according to the results obtained. It's a well-known fact that all AI Engineers have very unreliable cars, and although mine has proved very reliable, this is still quite a good place to start.

There are a very large number of problems, which can conspire to insure that you're not getting optimum performance from your car. For most of us, the most serious problem is when the car won't start, and so this should be the first consideration of our computerised car maintenance expert. The easiest way is if it asks:

CAR WON'T START?

An affirmative answer selects

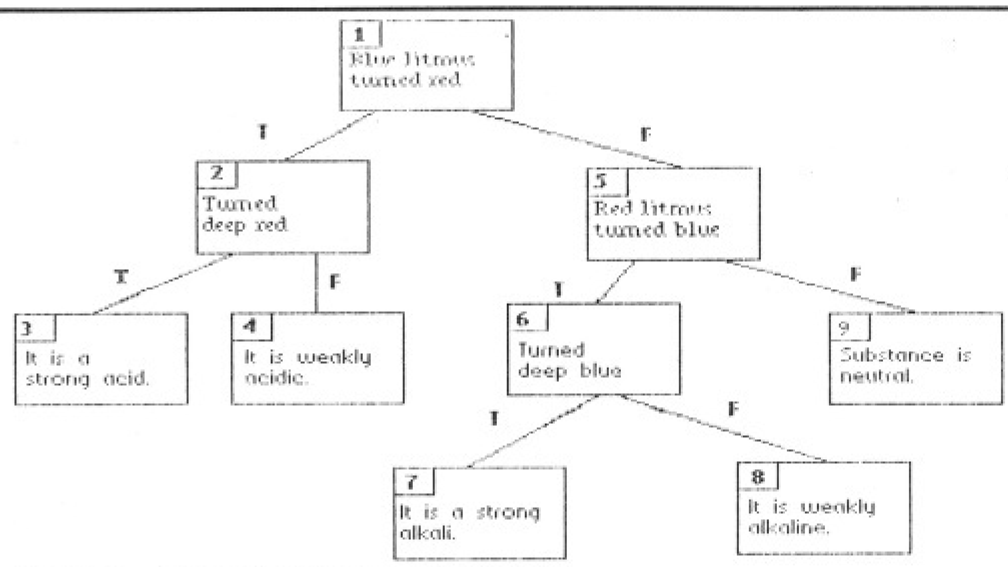


Figure 1 - Tree structure of the Example Knowledge Base

analysis of this problem, otherwise other problems are considered. If we consider just the case of the car not starting, we now investigate the simplest reasons first, so as to minimise the amount of work involved in isolating the cause of the problem. The next question is likely to be:

FUEL TANK IS EMPTY?

and if this is not the case,

BATTERY IS FLAT?

A negative response at this point will lead to a series of questions to trace through the fuel and electrical systems. Car maintenance is a good example of a deterministic expert system, because it clearly demonstrates the two big advantages of an expert system:

1. A guide through the steps to analyse a problem

2. Ultimately identifies a solution.

Fuzzy Logic

The part of this example we've looked at so far works very nicely, as it's quite simple to answer TRUE or FALSE to each question. If we proceed a little further, we will encounter the question:

IS SPARK GOOD ON ALL PLUGS?

We can easily tell whether there is a spark, but how do we know if it's adequate? Much research is currently being done into fuzzy logic. As there's no measurable divide between a good and inadequate spark, a probability is associated with the measured value within the critical range, and can be used in conjunction with probabilities for other components to give an assessment of whether the combination works.

At this stage we now have a non-deterministic expert system, which can't give a definite answer, just the most probable answer. As most home motor mechanics would have no means of accurately measuring the strength of the spark, these niceties can be ignored, and the somewhat imprecise question employed. This falls in rather nicely with the LISP list approach.

REASON 128

REASON 128 is an interactive Expert System tool which enables you to create, modify, save and interrogate a moderately sized knowledge base. It utilises the fact that almost all knowledge bases can be represented as a simple tree structure.

This representation greatly simplifies the problem of constructing a Knowledge Base, as each statement need only be associated with two pointers to the next statement to be displayed, according to whether or not the previous one was TRUE or FALSE. This is of course a doddle in Pascal or C, as it's just a matter of defining records to create a binary tree. It can also be achieved very simply in basic using three arrays.

Designing a Knowledge Base

The design of a Knowledge Base requires two things:

1. some knowledge of the subject (e.g text book).

2. a large sheet of paper.

The reference text provides the information on the question/tests and eventually the conclusions and the paper are used to arrange them in a tree structure.

Let's move on now out of the garage and into the chemistry lab. At the back of a shelf we find an unlabelled bottle containing a clear liquid. This could be water, sulphuric acid, caustic soda or countless other things, and it would be nice to know which one it is. To keep this example reasonably short, we'll only consider tests related to the acidity or alkalinity of the liquid.

We can do very simple tests using litmus to determine whether it's acidic or alkaline, and also make a subjective analysis of these tests. This can be defined in terms of four questions and five conclusions, with these arranged into a simple tree structure as shown in Figure 1. If we get a positive result to test 1, we can ignore the righthand half of the tree and proceed to question 2. The answer to this results in conclusion 3 or 4.

```
REASON 128 - QUESTIONS / CONCLUSIONS

[ 1 ]   BLUE LITMUS IS TURNED RED T->[ 2 ] F->[ 5 ]
[ 2 ]   IT TURNED DEEP RED T->[ 3 ] F->[ 4 ]
[ 3 ]   IT IS A STRONG ACID T->[ 0 ] F->[ 0 ]
[ 4 ]   IT IS WEAKLY ACIDIC T->[ 0 ] F->[ 0 ]
[ 5 ]   RED LITMUS IS TURNED BLUE T->[ 6 ] F->[ 9 ]
[ 6 ]   IT TURNED DEEP BLUE T->[ 7 ] F->[ 8 ]
[ 7 ]   IT IS A STRONG ALKALI T->[ 0 ] F->[ 0 ]
[ 8 ]   IT IS WEAKLY ALKALINE T->[ 0 ] F->[ 0 ]
[ 9 ]   IT IS NEUTRAL T->[ 0 ] F->[ 0 ]
```

Figure 1 : Chemistry Knowledge Base

Working with REASON 128

Once you have a tree diagram like the one in Figure 1, you're ready to use REASON 128. Run program one, and a list of six options is displayed. Select 1 to create a new Knowledge Base. This will then prompt for the first question to be asked. Type in the text string you wish to be displayed, and press RETURN. It then prompts for the next questions to be asked, according to whether the answer was TRUE or FALSE. When you reach the end of a branch, these are entered as 0 to indicate that this is a conclusion rather than a question.

When you've entered all statements on your tree, type to the next prompt and you will then be asked if you want a hard copy of the Knowledge Base. This is in the format shown in Figure 2, and is useful both as a check, and for future reference if you wish to make any changes. You're then returned to the main menu.

At this point it is advisable to use option 3 to save the Knowledge Base to disk. If you don't have a Formatted data disk, use option 5 first. when a Knowledge Base has been saved, it can be reloaded using option 2. It's important to note that the Save routine adds the extension ".KB", and this part of the filename should not be specified when using load.

Finally, option 4 is used to interrogate your Knowledge Base. Just keep answering T(rue) or F(alse) to the questions until REASON highlights its conclusion.

Problems and Limitations

For reasonably small applications, you'll find REASON 128 both efficient and easy to use. For more complex applications, however, it is often difficult to construct a complete tree at the outset. It can also mean that you have to use an identical question at several parts of the tree, thus wasting valuable data space.

For such applications, it would be much nicer to have a system that allowed you to define just the fact which you knew about possible solutions. Later in the article, we'll look at three programmes that create such a system. You'll still find REASON useful, as such systems frequently produce multiple solutions. REASON can help in developing the extensions to eliminate these and improving the efficiency of the data structures.

Playing Games

Although REASON was designed primarily for setting up and interrogating simple tree structures, it need not be limited to such applications. One quite interesting area is repetitive applications that can occur in many simple games. Figure 3 is a listing of the REASON knowledge base for a Noughts and Crosses system. Like many such games, the strategy is complicated only for the opening moves, after which it is simply a repetition of a very simple set of rules.

```
REASON 128 - QUESTIONS / CONCLUSIONS

[ 1 ]   PLAYING X'S (X GOES FIRST) T->[ 2 ] F->[ 21 ]
[ 2 ]   PUT X IN ANY CORNER - NEXT MOVE T->[ 3 ] F->[ 4 ]
[ 3 ]   O PUT IN ADJACENT CORNER OR CENTRE T->[ 5 ] F->[ 6 ]
[ 4 ]   IF YOU DON'T WANT MY HELP THEN BYEBYE T->[ 0 ] F->[ 0 ]
[ 5 ]   PUT X IN OPPOSITE CORNER TO FIRST X - NEXT MOVE T->[ 11 ] F->[ 4 ]
[ 6 ]   PUT X IN AN ADJACENT CORNER FORCING O'S TO BLOCK - NEXT MOVE T->[ 7 ] F->
[ 4 ]
[ 7 ]   O'S BLOCKED LINE T->[ 8 ] F->[ 9 ]
[ 8 ]   O'S FIRST MOVE WAS TO CORNER OPPOSITE FIRST X T->[ 10 ] F->[ 32 ]
[ 9 ]   PLACE X TO COMPLETE LINE - YOU WIN THANKS TO MY EXPERT ADVICE T->[ 0 ] F-
>[ 0 ]
[ 10 ]   PUT X IN LAST EMPTY CORNER - NEXT MOVE T->[ 9 ] F->[ 4 ]
[ 11 ]   CAN COMPLETE LINE OF X'S T->[ 9 ] F->[ 12 ]
[ 12 ]   O'S CAN COMPLETE LINE T->[ 15 ] F->[ 13 ]
[ 13 ]   CENTRE SQUARE IS VACANT T->[ 16 ] F->[ 14 ]
[ 14 ]   A CORNER IS VACANT T->[ 17 ] F->[ 18 ]
[ 15 ]   PLACE X TO BLOCK LINE OF O'S - NEXT MOVE T->[ 19 ] F->[ 4 ]
[ 16 ]   PUT X IN CENTRE - NEXT MOVE T->[ 19 ] F->[ 4 ]
```

```
[ 17 ]  PUT X IN VACANT CORNER - NEXT MOVE T->[ 19 ] F->[ 4 ]
[ 18 ]  PUT X IN ANY EMPTY SQUARE - NEXT MOVE T->[ 19 ] F->[ 4 ]
[ 19 ]  2 OR MORE SQUARES EMPTY T->[ 11 ] F->[ 20 ]
[ 20 ]  GAME DRAWN - I ONLY PROMISED NOT TO LOSE T->[ 0 ] F->[ 0 ]
[ 21 ]  FIRST X WAS PUT IN CENTRE T->[ 22 ] F->[ 23 ]
[ 22 ]  PUT O IN ANY CORNER - NEXT MOVE T->[ 26 ] F->[ 4 ]
[ 23 ]  PUT O IN CENTRE - NEXT MOVE T->[ 33 ] F->[ 4 ]
[ 24 ]  CAN COMPLETE A LINE OF O'S T->[ 25 ] F->[ 26 ]
[ 25 ]  PLACE O TO COMPLETE LINE - YOU WIN THANKS TO MY EXPERT ADVICE T->[ 0 ] F
->[ 0 ]
[ 26 ]  X'S CAN COMPLETE LINE T->[ 27 ] F->[ 28 ]
[ 27 ]  PLACE O TO BLOCK LINE - NEXT MOVE T->[ 31 ] F->[ 4 ]
[ 28 ]  A CORNER IS VACANT T->[ 29 ] F->[ 30 ]
[ 29 ]  PUT O IN CORNER CLOSEST TO MOST X'S - NEXT MOVE T->[ 31 ] F->[ 4 ]
[ 30 ]  PUT O IN ANY EMPTY SQUARE - NEXT MOVE T->[ 31 ] F->[ 4 ]
[ 31 ]  2 OR MORE SQUARES EMPTY T->[ 24 ] F->[ 20 ]
[ 32 ]  PUT X IN CENTRE - NEXT MOVE T->[ 9 ] F->[ 4 ]
[ 33 ]  TWO X'S IN OPPOSITE CORNERS T->[ 34 ] F->[ 26 ]
[ 34 ]  PUT O NEXT TO ONE OF THE X'S - NEXT MOVE T->[ 24 ] F->[ 4 ]
```

Figure 3 : Noughts and Crosses Knowledge Base

REASONING ON THE 128

PROGRAM ONE

```
10 GOTO 30000
999 REM GENERAL DISK ACCESS SUBR
OUTINE
1000 PRINT "CURRENT DISK IS :":P
RINT:SLOW:CATALOG:FAST
1010 PRINT "INSERT DATA DISK AND
 PRESS <SPACE> TO CONTINUE.":PRI
NT
1020 GETKEY Y$:IF Y$<>" "THEN 10
20
1030 F$="":PRINT "DATA DISK DIRE
CTORY :":PRINT:SLOW:CATALOG:FAST
:PRINT:INPUT"NAME OF KNOWLEDGE B
ASE (12 CHARS MAX) ":F$
1040 IF LEN(F$)>12 THEN F$=LEFT$
(F$,12)
1050 F$=F$+".KB"
1060 RETURN
10000 REM NEW KNOWLEDGE BASE
10010 PRINT"(RVS) DEFINE NEW KNO
WLEDGE BASE (OFF)":PRINT:PRINT"I
NITIALISING VARIABLES - PLEASE W
AIT A MOMENT":PRINT
10020 FOR N=1 TO KZ
10030 Q$(N)="":T%(N)=0:F%(N)=0
10040 NEXT N
10050 PRINT"(RVS) EDIT QUESTIONS
 / CONCLUSIONS - ENTER '@' TO EX
IT (OFF)":PRINT
10060 N=1
10070 DO
10080 PRINT "[";N;"] ? ";
10090 IF Q$(N)<>"" THEN PRINT Q$
(N);:FOR L=1 TO LEN(Q$(N)):PRINT
 CHR$(157);:NEXT
10100 PRINT CHR$(157);CHR$(157);
10110 INPUT Q$(N)
10120 IF Q$(N)="@" THEN 10150
10130 PRINT "IF TRUE GOTO [N] (0
 IF NONE)  ";T%(N);CHR$(157):CHR
$(157);:L=LEN(STR$(T%(N))):FOR K
=1 TO L:PRINT CHR$(157);:NEXT K:
INPUT T%(N)
10140 PRINT "IF FALSE GOTO [N] (
0 IF NONE)  ";F%(N);CHR$(157);CH
R$(157);:L=LEN(STR$(F%(N))):FOR
K=1 TO L:PRINT CHR$(157);:NEXT K
:INPUT F%(N)
10150 N=N+1
10160 LOOP UNTIL N>KZ OR Q$(N-1)
="@"
10170 NC=N-1:IF Q$(NC)="@" THEN
NC=NC-1
10180 PRINT:INPUT "DO YOU WISH T
O PRINTOUT LIST OF CONDITIONS [Y
/N] ";Y$
10190 IF Y$<>"Y" AND Y$<>"|" THE
N 10230
10200 SLOW:OPEN 4,4:PRINT#4,"REA
SON 128 - QUESTIONS / CONCLUSION
S":PRINT#4
10210 FOR N=1 TO NC:PRINT#4,"[";
N;"]  ";Q$(N);" T->[";T%(N);"] F
->[";F%(N);"]":NEXT
10220 CLOSE 4:FAST
10230 WINDOW 0,3,79,24,1
10240 RETURN
15000 REM QUESTION KNOWLEDGE BAS
E
15010 PRINT"(BLK)(RVS)        ANS
WER QUESTIONS WITH (OFF)T(RVS)RU
E OR (OFF)F(RVS)ALSE UNTIL CONCL
USION IS REACHED        (OFF)(BLU)
";:WINDOW 0,5,79,24,1
15030 IF NC<>0 THEN 15060
15040 PRINT "(RED)KNOWLEDGE BASE
 NOT LOADED - PRESS ANY KEY FOR
MAIN MENU.(BLU)"
15050 GETKEY Y$:WINDOW 0,3,79,24
,1:RETURN
15060 N=1:C%=0
15070 DO
15080 IF T%(N)=0 AND F%(N)=0 THE
N C%=1:PRINT:PRINT"(BLK)(RVS)";
15090 PRINT Q$(N);:IF C%=1 THEN
PRINT"(BLU)(OFF)"
15100 IF C%=0 THEN BEGIN
15110 PRINT" [T/F] : ";
15120 GETKEY Y$:IF Y$<>"T" AND Y
$<>"F" AND Y$<>"|" AND Y$<>"-" T
HEN 15120
15130 PRINT Y$:IF Y$="T" OR Y$="
|" THEN 15160
15140 IF F%(N)<>0 THEN N=F%(N):G
OTO 15180
15150 PRINT"(RED)KNOWLEDGE BASE
INCOMPLETE !!":C%=1:GOTO 15180
15160 IF T%(N)<>0 THEN N=T%(N):G
OTO 15180
15170 PRINT"(RED)KNOWLEDGE BASE
INCOMPLETE !!":C%=1
15180 BEND
15190 LOOP UNTIL C%=1
15200 PRINT:PRINT"PRESS ANY KEY
FOR MAIN MENU.":GETKEY Y$
15210 WINDOW 0,3,79,24,1:SCNCLR:
RETURN
20000 REM LOAD KNOWLEDGE BASE
20010 PRINT"(RVS) LOAD KNOWLEDGE
 BASE (OFF)":PRINT:INPUT"ARE YOU
 SURE [Y/N] ";Y$:PRINT
20020 IF Y$<>"Y" AND Y$<>"|" THE
N RETURN
20030 GOSUB 1000
20040 SLOW:DOPEN#1,""+F$,R
20050 IF DS=0 THEN 20080
20060 PRINT "(RED)(OFF) DISK ERR
OR (BLU)(OFF) ";DS$
20070 PRINT "PRESS ANY KEY FOR M
AIN MENU.":GETKEY Y$:RETURN
20080 INPUT#1,NC
20090 FOR N=1 TO NC
20100 INPUT#1,Q$(N):INPUT#1,T%(N
):INPUT#1,F%(N)
20110 NEXT N:DCLOSE#1:FAST
20120 SCNCLR:PRINT "(RVS) KNOWLE
DGE BASE LOADED SUCCESSFULLY (OF
F)"
20130 CHAR 1,10,4,"1. EDIT KNOWL
EDGE BASE"
20140 CHAR 1,10,7,"2. RETURN TO
MAIN MENU"
20150 CHAR 1,5,10,"ENTER SELECTI
ON >> "
20160 GETKEY Y$:LM%=ASC(Y$)-48:I
F LM%<1 OR LM%>2 THEN 20160
20170 SCNCLR:IF LM%=1 THEN 10050
20180 RETURN
24000 REM SAVE
24010 PRINT"(RVS) SAVE KNOWLEDGE
 BASE (OFF)":PRINT:INPUT"ARE YOU
 SURE [Y/N] ";Y$:PRINT
24020 IF Y$<>"Y" AND Y$<>"|" THE
N RETURN
24030 GOSUB 1000
24040 SLOW:DOPEN#1,""+F$,D0,U8,W
:IF DS=0 THEN EX%=0:GOTO 25000
24050 IF DS=63 THEN DCLOSE#1:GOT
```

```
O 24090
24060 PRINT "(RED)(RVS) DISK ERR
OR (BLU)(OFF) ";DS$:DCLOSE#1
24070 FAST:PRINT "PRESS ANY KEY
FOR MAIN MENU."
24080 GETKEY Y$:RETURN
24090 EX%=1:INPUT "(RVS) KNOWLED
GE BASE EXISTS (OFF) - REPLACE [
Y/N] ";Y$
24100 IF Y$<>"Y" AND Y$<>"|" THE
N FAST:RETURN
24110 DOPEN#1,"@"+F$,D0,U8,W:IF
DS<>0 THEN 24060
25000 REM OK TO SAVE, BUT IS THE
RE ANY DATA
25010 IF NC<>0 THEN 25040
25020 PRINT "(RED)KNOWLEDGE BASE
 EMPTY(BLU) - PRESS ANY KEY FOR
MAIN MENU."
25030 GETKEY Y$:DCLOSE#1:IF EX%=
0 THEN SCRATCH ""+F$:FAST:RETURN
:ELSE FAST:RETURN
25040 REM SAVE QUESTIONS
25050 PRINT#1,NC
25060 FOR N=1 TO NC
25070 PRINT#1,Q$(N):PRINT#1,T%(N
):PRINT#1,F%(N)
25080 NEXT N:DCLOSE#1:FAST
25090 PRINT "KNOWLEDGE BASE SAVE
D - PRESS ANY KEY FOR MAIN MENU.
"
25100 GETKEY Y$:RETURN
28000 REM FORMAT
28010 PRINT"(RED)(RVS) FORMAT DA
TA DISK (OFF)(BLU)":PRINT:INPUT"
ARE YOU SURE [Y/N] ";Y$:PRINT
28020 IF Y$<>"Y" AND Y$<>"|" THE
N RETURN
28030 PRINT"INSERT BLANK DISK IN
 DRIVE."
28040 INPUT"ENTER DISK NAME [16
CHARS MAX] ";H$
28050 IF LEN(H$)>16 THEN H$=LEFT
$(H$,16)
28060 SLOW:H$="N0:"+H$+",KB":OPE
N 1,8,15,H$:CLOSE 1:IF DS=0 THEN
 PRINT"(BLK)":CATALOG:PRINT"(BLU
)":ELSE PRINT"(RED)(RVS)";DS$;"(
BLU)(OFF)":FAST
28070 PRINT:INPUT"ANOTHER DISK [
Y/N] ";Y$
28080 IF Y$="Y" OR Y$="|" THEN P
RINT:GOTO 28030
29999 RETURN
30000 REM START OF MAIN PROGRAM
30010 GRAPHIC CLR:REM GET EXTRA
9K
30020 GRAPHIC 0,1:FAST:COLOR4,16
:COLOR0,7:COLOR5,16
30030 CHAR 1,6,10,"
                    ",1
30040 CHAR 1,6,11," SWITCH TO 80
 COLUMN SCREEN ",1
30050 CHAR 1,6,12,"
                    ",1
30060 GRAPHIC 5,1:REM SELECT 80
COLUMN SCREEN
30070 COLOR 5,1:COLOR 6,16
30080 WINDOW 0,0,79,2,1
30090 PRINT "      ─┬┬──────────
─┬┬┬─────────────────────────┬─
┬──────────────────────┐"
30100 PRINT "      |||(RED) REASO
N 128(BLK) |||(BLU) A KNOWLEDGE
BASE PROGRAM(BLK) |||(RED) BY PA
UL SCHOFIELD(BLK) |||"
30110 PRINT " \     ┴┴──────────
─┴┴┴─────────────────────────┴─
┴──────────────────────┘";
30120 COLOR5,7:WINDOW 0,3,79,24,
1:GOSUB 40000
30130 CHAR 1,20,3,"
                          ",1
30140 CHAR 1,20,4,"
    MENU                  ",1
30150 CHAR 1,20,5,"
    ====                  ",1
30160 CHAR 1,20,6,"
                          ",1
30170 CHAR 1,20,7,"    1 )  DEFI
NE KNOWLEDGE BASE         ",1
30180 CHAR 1,20,8,"
                          ",1
30190 CHAR 1,20,9,"    2 )  LOAD
 KNOWLEDGE BASE           ",1
30200 CHAR 1,20,10,"
                          ",1
30210 CHAR 1,20,11,"    3 )  SAV
E KNOWLEDGE BASE          ",1
30220 CHAR 1,20,12,"
                          ",1
30230 CHAR 1,20,13,"    4 )  QUE
STION KNOWLEDGE BASE      ",1
30240 CHAR 1,20,14,"
                          ",1
30250 CHAR 1,20,15,"    5 )  FOR
MAT DATA DISK             ",1
30260 CHAR 1,20,16,"
                          ",1
30270 CHAR 1,20,17,"    6 )  EXI
T                         ",1
30280 CHAR 1,20,18,"
                          ",1
35000 GETKEY I$
35010 I%=VAL(I$)
35020 IF I%<1 OR I%>6 THEN 35000
 :ELSE SCNCLR
35030 ON I% GOSUB 10000,20000,24
000,15000,28000,39999
35040 SCNCLR:GOTO 30130
39999 WINDOW 0,0,79,24,1:END
40000 REM DIMENSION ARRAYS ONLY
ONCE
40010 KZ=500
40020 DIM Q$(KZ),T%(KZ),F%(KZ)
40030 RETURN
```

THE ULTIMATE BACKUP CARTRIDGE GOES ONE BETTER...

NOW ACTION REPLAY Mk V PROFESSIONAL

FOR CBM64/128 HAS ARRIVED



EVEN MORE POWERFUL, MORE FRIENDLY & NOW EVEN MORE FEATURES!

ALL FEATURES AVAILABLE TO TAPE OR DISK USERS.

ACTION REPLAY Mk V differs from all other cartridges because it combines an 8K RAM with a FULL 32K operating system ROM. That means that ALL features are INSTANTLY AVAILABLE AT ALL TIMES.

WARNING!! Other systems use outdated technology which severly limits performance. Action Replay's state of the art hardware gives you MORE POWER, MORE SPEED, MORE FACILITIES than any other cartridge. There really is no comparison. Here are just some of the features...

✓ **TURBO RELOAD** Action Replay Mk V has 2 unique Turbo Loaders - "Ramloader" & "Warp 25". Both work at up to 25 times normal speed! ✓ **PRINTER DUMP** freeze any game & print out the screen. ✓ **PICTURE SAVE** save any HiRes multi colour screen to disk at the push of a button. ✓ **SPRITE CONTROL** full range of controls with 64K operation. ✓ **POKEFINDER GENERAL** an automatic infinite lives generator! ✓ **SUPER COMPACTOR** ultra efficient program compaction techniques. ✓ **TEXT SCREEN EDITOR** modify the text screen on a frozen program. ✓ **SUPERFAST DISK OPERATION** load 200 blocks in just SIX SECONDS. ✓ **MORE TAPE FACILITIES** you don't need a disk drive to use Action Replay. ✓ **TOOLKIT COMMANDS** full range all at the press of a button. ✓ **FULLY INTEGRATED OPERATION** with onboard custom LSI LOGIC PROCESSING CHIP. ✓ **CENTRONICS INTERFACE** for parallel printers. ✓ **PROFESSIONAL MACHINE CODE MONITOR** Full 64K monitor available at all times with full range of commands.

PERFORMANCE PROMISE

Action Replay has an unmatched range of onboard features. When you buy Action Replay, if you don't find our claims to be true, then return it within 14 days for a full refund.

GRAPHICS SUPPORT UTILITIES DISK

SLIDE SHOW. View your favourite screens in a slide show type display.

BLOW UP. A unique utility to allow you to take any part of of your picture & 'blow it up 'to full screen size.

SPRITE EDITOR. A complete sprite editor helps you to create or edit sprites.

MESSAGE MAKER. Takes your favourite screen created with a graphics package or captured with Action Replay & turns it into a scrolling screen message complete with music.

ONLY £12.99

NOTICE 1988 COPYRIGHT ACT

DATEL ELECTRONICS Ltd. neither authorizes or condones the use of it's products to reproduce copyright material. It is illegal to make copies of such material without the expressed consent of the copyright owners or thier licencees.

TOTAL BACKUP POWER

The most powerfull backup system ever devised. Unstoppable freezer system. Freeze at any point. Just press the magic button to backup your programs.

❑ TAPE TO TAPE ❑ TAPE TO DISK ❑
❑ DISK TO DISK ❑ DISK TO TAPE ❑

VERY fast & efficient program compaction. Single part save at TURBO speed to disk or tape. Backups turboload INDEPENDENTLY of cartridge. No "programing" or user knowledge required.

THE PROCESS IS AUTOMATIC - JUST GIVE THE BACKUP A NAME.

WHAT THE REVIEWERS SAID

"I'm stunned, amazed and totally impressed. This is easily the best value for money cartridge. *The Cartridge King!"*

Commodore Disk User

UPGRADE INFORMATION

Mk IV Professional to Mk V Professional - just send £9.99 & we will send you a new Mk V Operating System Chip for you to just plug in! Mk IV (Standard) to Mk V Professional - send your old cartridge plus £15.99 & we will upgrade it (allow 14 days).

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0782 744707 24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	0782 744292 UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLERS WELCOME - Please reserve goods by telephone prior to visit.

DATEL ELECTRONICS

DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.

SALES ONLY 0782 744707

TECHNICAL ONLY 0782 744324

Thesaurus

Have you ever been stuck for words while writing a letter or document? Then perhaps THESAURUS can help you

By Norman Hart

As you may have guessed, this program's function is to present groups of words with similar meanings, derived from an alphabetically presented list. All procedures are menu driven, so even if you don't possess great typing skills, mistaken entries are impossible.

The database is sorted out in such a way that searches are unnecessary; the results of your enquiries are derived directly from the array in memory, and are displayed instantly on the screen – time after time. The program is of course crash-proof, with [RUN/STOP-RESTORE] your only means of escape.

How it works

The program begins by POKEing an alphabetical sort into location 49152 (the value of A$ in line 180 – this can be changed if you wish), the data for this is held in lines 200 to 320. By the way, do take care with the values, since the sort is a *very* essential procedure.

The main database (lines 1860 onwards) is then sized up and dimensioned (lines 360 & 370) before being committed to memory as the A$() array. This first array is a simple representation of the database, word for word, and acts as the resource for your enquiries. In order for this pool of information to be tapped, a controlling array is required – namely the L$() array.

This second array is created on the screen, for example "ABATE 120:" is A$(120), together with its own subscript, is Input into memory as part of the L$() array – see INPUT 1, L$(J) in line 460. Thus we eventually end up with two arrays – the A$(), representing the database, and L$(), representing the A$(), together with the appropriate A$() subscripts MINUS the asterisks. Check line 360 for E=E+1 to see how I counted out the asterisks in the DIM L$ (D-E-1) in line 380, and check line 400 to see how I avoided their appearance on the screen in line 450.

Once the L$() array is completed, the SYS AS in line 420 sorts the array into alphabetical order. Despite its machine code pedigree, the sort has a lot of work to do, so be prepared for a wait of up to three minutes in this large program example! With that important job done, the next task is to create a special Control String that allows the user to call up the word list at whatever point he or she wishes on the screen.

This string (C$) is a compilation of array position numbers for each fresh alphabetical change that occurs in the L$() array. In other words, if the user presses the letter "D", the list on the screen will begin with words starting with the letter "D". This facility works for the entire alphabetical range. The SPACE bar also allows the list to be paged forward and back in follow-on order. In this way words can be accessed very quickly. Study lines 500 to 560 to see how the Control String (C$) is created.



Now that the program has initiated itself, an alphabetical list appears on the screen – this always begins with words starting with the letter "A". This list appears alongside all the necessary prompts and safe guards. All you have to do is press the appropriate letter key and, if needed, the SPACE bar (Shift/SPACE if you wish to page backwards), then press the RETURN key and the top left-hand word will appear in Reverse.

Now use the up and down CURSOR key to position the Reverse hightlight over the word you wish to investigate. Press the RETURN key again, and your list of associated words will appear on the screen. For example, the word BIG will reveal such words as GIANT, MASSIVE, ENORMOUS, etc.. The F1 key allows you to return to the same previous position in the main list.

Print Outs

Not only is information available on the screen, but the printer has its share of responsibilities, and there are three types of print out at your disposal:

1. The full Data Base as it appears at the end of the program. Tap the F2 key, then immediately hold down the CTRL key until the printer starts.
2. The Alphabetical List together with the appropriate A$() subscripts. This print out begins from its alphabetical appearance on the screen. Tap the F4 key, then immediately hold down the CTRL key until the printer starts.
3. A print out of your selected group of words. Press the F2 key only.

Print items 1 & 2 are only available while the Alphabetical List is on the screen. They are protected from casual use by employing the CTRL key as described, and also there are a lack of prompts on the screen – I have intended their access only to you, the programmer. Print item 3 is only accessible while your selected group of words appears on the screen. All print outs end with a word count, and printing can be abandoned by holding down any key on the C 64. Of course if you don't have a printer, simple REM out lines 780, 790 & 1410.

As for the databse itself, it need not be restricted to a Thesaurus. Any groups of words, ideas or items that share a common interest can be included – parts for equipment perhaps, or even foreign language translation. Just remember to encompass them by asterisks as I have done in the enclosed example, and finish the data statements with a double asterisk "**"

Not only will you find this program useful for your own literary efforts, but any younger members of the family will certainly benefit too. My eight year old son, Ian, told me to tell you!

Listings

PROGRAM: THESAURUS

```
AF   10 REM****************
63   20 REM*    CBM 64     *
C5   30 REM*               *
2B   40 REM*  THESAURUS    *
F1   50 REM*               *
ED   60 REM* WRITTEN BY    *
FD   70 REM*               *
DB   80 REM* NORMAN HART   *
FF   90 REM****************
3A   100 POKE788,52:REM:RUN/STOP
     DISABLE
81   110 FORX=1TO39:S$=S$+CHR$(32
     ):NEXT:PRINTFRE(8)
DF   120 PRINTCHR$(147)CHR$(144)C
     HR$(142)CHR$(8):POKE53281,15
     :OPEN1,3:OPEN2,4
D7   130 C=13:L=8:GOSUB1450:PRINT
     "*THESAURUS*"
0F   140 L=10:GOSUB1450:PRINT"WRI
     TTEN  BY"
6D   150 L=12:GOSUB1450:PRINT"NOR
     MAN HART"
1D   160 L=14:GOSUB1450:PRINT"*JA
     N  1989*
90   170 :
F5   180 AS=49152:REM LOC OF SORT
B4   190 :
B8   200 DATA32,115,0,133,97,169,
     128,133,98,32,115,0,240,7,9,
     128,133,98,32,115
B1   210 DATA0,165,47,133,99,165,
     48,133,100,160,0,165,97,209,
     99,208,7,200,165,98
9E   220 DATA209,99,240,20,24,160
     ,2,177,99,101,99,72,200,177,
     99,101,100,133
B7   230 DATA100,104,133,99,144,2
     21,160,5,177,99,133,102,200,
     177,99,133,101,208
E7   240 DATA2,198,102,198,101,24
     ,165,99,105,7,133,99,165,100
     ,105,0,133,100,165,101
3C   250 DATA208,2,198,102,198,10
     1,208,4,165,102,240,18,133,1
     05,162,0,134,103,134
30   260 DATA104,165,99,133,106,1
     65,100,133,107,240,224,240,1
     14,24,165,106,105
B4   270 DATA3,133,106,165,107,10
     5,0,133,107,230,103,208,2,23
     0,104,160,2,177,106
D9   280 DATA153,109,0,136,16,248
     ,160,5,177,106,153,109,0,136
     ,192,2,208,246,170
BD   290 DATA56,229,109,144,2,166
     ,109,160,255,232,200,202,208
     ,8,165,112,197,109
9D   300 DATA144,10,176,34,177,11
     3,209,110,240,238,16,26,160,
     2,185,112,0,145
FF   310 DATA106,136,16,248,160,5
     ,185,106,0,145,106,136,192,2
     ,208,246,169,0,133
9A   320 DATA105,165,101,197,103,
     208,152,165,102,197,104,208,
     146,165,105,240,138,96
71   330 :
6F   340 :
ED   350 FORJ=ASTOAS+242:READX:PO
     KEJ,X:POKE53280,X:NEXT
C2   360 D=D+1:READA$:IFA$="*"THE
     NE=E+1
FD   370 IFA$<>"**"THENPOKE53280,
     ASC(A$):GOTO360
9E   380 DIMA$(D),L$(D-E-1):RESTO
     RE:FORJ=0TO242:READX:NEXT:PO
     KE53280,12:POKE53281,15
6F   390 J=0:L=20:GOSUB1450:PRINT
     D-E"WORDS":L=22:GOSUB1450:PR
     INTE"GROUPS"
EB   400 I=I+1:READA$(I):IFA$(I)=
     "*"THEN400
4B   410 IFA$(I)<>"**"THENGOSUB43
     0:GOTO400
C9   420 IFA$(I)="**"THENPRINT"AL
     PHABETICAL SORT":SYSAS,L:A$(
     0)="*":GOTO500
54   430 C=0:L=0:GOSUB1450:PRINTC
     HR$(155)LEFT$(S$,LEN(A$(I))+
     LEN(STR$(I))+4)
A0   440 C=LEN(A$(I))+4-LEN(STR$(
     I))
6B   450 GOSUB1450:PRINTSTR$(I)":
     ":C=0:GOSUB1450:PRINTA$(I)
03   460 GOSUB1450:J=J+1:INPUT#1,
     L$(J)
```

```
66  470 PRINTLEFT$(S$,5)CHR$(145)CHR$(144):PRINTD-E-J:RETURN
DB  480 :
D1  490 :
2E  500 L$(0)="*":PRINTCHR$(147);
9B  510 FORX=1TOD-E-1:IFASC(L$(X))<>ASC(L$(X-1))THENGOSUB540:POKE53280,Y
44  520 NEXT:PRINT":"CHR$(19);
C9  530 INPUT#1,B$:INPUT#1,C$:C$=B$+C$+CHR$(32)+CHR$(160)+"A":GOTO590
38  540 PRINTCHR$(155)LEFT$(L$(X),1)RIGHT$(STR$(X),LEN(STR$(X))-1);
BE  550 Y=Y+1:IFY=20THENY=0:PRINT":"
BC  560 RETURN
06  570 :
7C  580 :
C0  590 PRINTCHR$(147)CHR$(144):POKE53280,15:H=65
07  600 PRINT:PRINT:PRINT"[CA,S*18,CR,S*18,CS]"
78  610 FORX=0TO18:PRINT"[S-]"SPC(18)"[S-]"SPC(18)"[S-]":NEXT
FA  620 PRINT"[CZ,S*18,CE,S*18,CX]"
4B  630 GOTO820
8C  640 X=0:PRINTCHR$(19)CHR$(146)"KEY A-Z=LIST[SPC11][RETURN]=ENQUIRE"
63  650 PRINT"(SHIFT)/[SPACE]=PAGE "
F3  660 PRINT"LAST ENQUIRY :";
0B  670 IFH=133THENPRINTLEFT$(L$(Q+V),LEN(L$(Q+V))-4)LEFT$(S$,22-LEN(L$(Q+V)))
90  680 :
8E  690 :
E7  700 POKE198,0
F8  710 IFPEEK(198)=0ANDX<10THENX=X+1:GOTO710
DE  720 PRINTCHR$(19)CHR$(18)SPC(4)"A-Z"SPC(17)"RETURN"
44  730 PRINTCHR$(18)SPC(1)"SHIFT"SPC(3)"SPACE"
06  740 IFPEEK(198)=0ANDX<60THENX=X+1:GOTO740
55  750 IFPEEK(198)=0THEN640
22  760 H=PEEK(631):IFH>47ANDH<58THEN640
32  770 IFH=13THENC=1:L=4:V=0:PRINTCHR$(18);:GOTO1000
B5  780 IFH=137THENFORX=1TO1000:NEXT:IFPEEK(653)=4THENGOSUB1520
AA  790 IFH=138THENFORX=1TO1000:NEXT:IFPEEK(653)=4THENGOSUB1690
0E  800 IFH=32ANDR>D-E-1THEN640
31  810 IFH=160ANDQ<2THEN640
D1  820 L=4:GOSUB1450:T=1
B2  830 FORX=1TOLEN(C$)
92  840 IFCHR$(H)=MID$(C$,X,1)THENY=X+1:GOTO870
C2  850 IFH<>133THENNEXT
6B  860 GOTO640
F2  870 IFMID$(C$,Y,1)<"A"THENY=Y+1:GOTO870
4A  880 IFH=32THENQ=Q+38:S=0:GOTO940
6E  890 IFH=160THENQ=Q-38:S=0:GOTO910
B8  900 IFH<>133THENQ=VAL(MID$(C$,X+1,Y-X-1))
7E  910 IFQ<1THENQ=1
A1  920 :
9F  930 :
B9  940 R=Q+37:S=0
81  950 FORP=QTOR
A5  960 IFP=Q+19THENT=20:GOSUB1450
21  970 PRINTTAB(T)LEFT$(S$,18)
3E  980 IFP<D-ETHENPRINTTAB(T)CHR$(145)LEFT$(L$(P),LEN(L$(P))-4):S=S+1
00  990 NEXT:X=LEN(C$):GOTO850
C2  1000 POKE198,0:IFV=19ANDH=17THENC=20:L=4
97  1010 IFV=SANDH=17THENV=0:C=1:L=4
97  1020 IFV=-1ANDH=145ANDS>18THENV=S-1:C=20:L=S-16
E3  1030 IFV=-1ANDH=145ANDS<19THENV=S-1:C=1:L=S+3
04  1040 IFV=18ANDH=145THEN:C=1:L=22
98  1050 GOSUB1480:IFH<>0THEN1140
12  1060 :
08  1070 :
5E  1080 X=0:PRINTCHR$(19);:PRINT"[F1]=RESTART[SPC11][RETURN]=ENQUIRE"
93  1090 PRINT"MOVE [CRSR] UP & DOWN"
7A  1100 IFPEEK(198)=0ANDX<10THENX=X+1:GOTO1100
99  1110 PRINTCHR$(19)CHR$(18)SPC(1)"F1"SPC(21)"RETURN"
B3  1120 PRINTCHR$(18)SPC(6)"CRSR"
CE  1130 IFPEEK(198)=0ANDX<60THENX=X+1:GOTO1130
A6  1140 IFPEEK(198)=0THEN1080
5B  1150 H=PEEK(631)
0E  1160 GOSUB1480
C5  1170 IFH=17THENV=V+1:L=L+1:PRINTCHR$(18);:GOTO1000
89  1180 IFH=145THENV=V-1:L=L-1:PRINTCHR$(18);:GOTO1000
D3  1190 IFH=133THENH=0:L=4:T=1:C=0:V=0:PRINTCHR$(19):GOTO640
51  1200 IFH=13THENL=4:I=VAL(RIGHT$(L$(Q+V),4)):GOTO1220
21  1210 PRINTCHR$(18);:POKE198,0:GOTO1050
88  1220 PRINTCHR$(19);:FORX=1TO3:PRINTS$:NEXT:PRINT
05  1230 FORX=0TO18:PRINTTAB(1)CHR$(18)LEFT$(S$,18)SPC(1)LEFT$(S$,18):NEXT
65  1240 IFA$(I)<>"*"THENI=I-1:GOTO1240
24  1250 IP=I+1:C=1:L=4:GOSUB1450:S=0:T=1
F9  1260 I=I+1:PRINTTAB(T)CHR$(18)A$(I):S=S+1:IFS=19THENT=20:GOSUB1450
0D  1270 IFA$(I+1)<>"*"THEN1260
3F  1280 :
35  1290 :
47  1300 X=0:PRINTCHR$(19)CHR$(146)."[F1]=RETURN[SPC4][F2]=PRINT OUT"
BA  1310 PRINT
5E  1320 PRINT"THIS ENQUIRY :";
CF  1330 PRINTLEFT$(L$(Q+V),LEN(L$(Q+V))-4)LEFT$(S$,22-LEN(L$(Q+V)))
8A  1340 POKE198,0
D6  1350 IFPEEK(198)=0ANDX<10THENX=X+1:GOTO1350
75  1360 PRINTCHR$(19)CHR$(18)SPC(1)"F1"SPC(13)"F2"
39  1370 IFPEEK(198)=0ANDX<60THENX=X+1:GOTO1370
82  1380 IFPEEK(198)=0THEN1300
CE  1390 H=PEEK(631)
4C  1400 IFH=133THENGOSUB1450:T=1:S=0:GOTO950
0F  1410 IFH=137THENGOSUB1610
9A  1420 GOTO1300
A1  1430 :
9F  1440 :
2E  1450 POKE211,C:POKE214,L:SYS58732:RETURN
83  1460 :
F9  1470 :
74  1480 GOSUB1450:PRINTLEFT$(L$(Q+V),LEN(L$(Q+V))-4)LEFT$(S$,22-LEN(L$(Q+V)))
59  1490 RETURN
DB  1500 :
D1  1510 :
8B  1520 GOSUB1810:X=0:Y=0
09  1530 PRINT#2,CHR$(13)"ARRAY PRINT OUT"CHR$(13)"---------------------"
8A  1540 X=X+1:IFA$(X)<>"*"THENW=W+1:PRINT#2,A$(X)LEFT$(S$,25-LEN(A$(X)));:Y=Y+1
41  1550 IFY=3THENPRINT#2,"":Y=0
F1  1560 IFA$(X)="*"THENPRINT#2,CHR$(13):Y=0:IFPEEK(197)<>64THENW=W+1:GOTO1580
DD  1570 IFA$(X)<>"**"THENGOTO1540
18  1580 PRINT#2,CHR$(13)"END OF LIST":PRINT#2:GOTO1770
06  1590 :
7C  1600 :
16  1610 GOSUB1810:Y=0:X=0:PRINT#2,CHR$(13)
1E  1620 PRINT#2,"ENQUIRY-"LEFT$(L$(Q+V),LEN(L$(Q+V))-4)CHR$(13)
18  1630 W=W+1:PRINT#2,A$(IP)LEFT$(S$,25-LEN(A$(IP)));:X=X+1:Y=Y+1
86  1640 IFY=3THENPRINT#2,"":Y=0:IFPEEK(197)<>64THENW=W+1:GOTO1660
4D  1650 IFA$(IP)<>"*"THENIP=IP+1:GOTO1630
50  1660 PRINT#2:IP=IP-X+1:GOTO1770
B6  1670 :
AC  1680 :
4E  1690 GOSUB1810:X=0:Y=0
B3  1700 PRINT#2,CHR$(13)"ALPHABETICAL PRINT OUT"CHR$(13)"-----------------------"
B1  1710 FORX=QTOD-E-1:W=W+1:PRINT#2,L$(X)LEFT$(S$,25-LEN(L$(X)));:Y=Y+1
70  1720 IFY=3THENPRINT#2,"":Y=0:IFPEEK(197)<>64THENX=D-E-1:W=W+1
4A  1730 NEXT
E8  1740 :
E6  1750 :
BE  1760 PRINT#2,CHR$(13)"END OF LIST":PRINT#2
1C  1770 PRINT#2,"NUMBER OF WORDS="W-1:PRINT#2:W=0
A7  1780 PRINTCHR$(19);:FORX=1TO3:PRINTS$:NEXT:RETURN
3E  1790 :
35  1800 :
A0  1810 PRINTCHR$(19);:FORX=1TO
```

```
   3:PRINTS$:NEXT
CF 1820 PRINTCHR$(19):PRINTTAB(
   8)"ERROR! >PRINTER OFF LINE"
   :PRINT#2
7F 1830 PRINTTAB(3)CHR$(145)"HO
   LD DOWN ANY KEY TO ABORT PRI
   NTING":RETURN
0D 1840 :
7B 1850 :
55 1860 DATABIG,GIANT,MASSIVE,E
   NORMOUS,COLOSSAL,GIGANTIC,TR
   EMENDOUS(SIZE),TITANIC
F1 1870 DATACAVERNOUS,GREAT(SIZ
   E),*
27 1880 DATAFIRE(HOT),ABLAZE,BL
   AZING,CONFLAGRATION
D0 1890 DATAGLOWING(HEAT),RED H
   OT,*
50 1900 DATASAD,UNHAPPY,UPSET,D
   EPRESSED,DEVASTATED,*
83 1910 DATAABDICATE,RENOUNCE,G
   IVE UP,FINISH WITH,*
04 1920 DATAFRIGHTENED,SCARED,W
   ORRIED,NERVOUS,TERRIFIED,HOR
   RIFIED,*
A2 1930 DATAABORT,REJECT,THROW
   AWAY,GIVE AWAY,SHED,RID,CAST
    OUT
65 1940 DATAABANDON,GIVE UP,GIV
   E IN,FORSAKE,YIELD,*
AB 1950 DATAKILL,MURDER,EXECUTE
   (KILL),DESTROY,ANIHALATE,EXT
   ERMINATE,OBLITERATE
F1 1960 DATAASSASINATE,BUTCHER,
   ERRADICATE,*
27 1970 DATAHIGH,ABOVE,BEYOND R
   EACH,OVERHEAD,*
33 1980 DATACARRY OUT,EXECUTE(C
   ARRY OUT),INITIATE,START,BEG
   IN,IMPLEMENT,*
91 1990 DATAGOOD,SUPERB,FANTAST
   IC(GOOD),SUPER,BRILLIANT
6A 2000 DATAZENITH,FINEST,PINNA
   CLE(BEST),CLIMAX
7F 2010 DATACULMINATION,TREMEND
   OUS(IDEA),SUPERLATIVE
B1 2020 DATAULTIMATE,OUTSTANDIN
   G,EXTRAORDINARY
94 2030 DATAUNEQUALLED,SPLENDID
   ,SUPERIOR,PERFECT,MARVELOUS,
   FAULTLESS
42 2040 DATAMATCHLESS,EXEMPLARY
   ,IMPECCABLE,*
D9 2050 DATADEAD,LIFELESS,DECEA
   SED,DEFUNCT,INANIMATE,PERISH
   ED
60 2060 DATASTIFF,CORPSE,CADAVE
   R,CARCASS,REMAINS(HUMAN),*
64 2070 DATAEMBARRASS,ABASH,HUM
   ILIATE,DISCONCERT,*
2F 2080 DATAABSENT,GONE,NOT PRE
   SENT,NON EXISTANT,FORGETFUL,
   INATTENTIVE,*
60 2090 DATAABATE,DIMINISH,SMAL
   L,LESS,WEAKEN,TINY,DIMINUTIV
   E
11 2100 DATAABREVIATE,SHORTEN,C
   UT DOWN
C3 2110 DATAMICROSCOPIC,PETITE,
   *
E6 2120 DATAABSORB,TAKE IN,INGE
   ST,DIGEST,CONSUME,*
8E 2130 DATAFAILURE,MALFUNCTION
   ,BREAKDOWN,DESTRUCTION,*
9C 2140 DATAERROR,MISTAKE,BLOOM
   ER,FAULT,HOWLER,INACCURACY,M
   ISCALCULATION,*
E6 2150 DATAILLNESS,DISEASE,FEV
   ER,PAIN,ACHE,SORE,TENDERNESS
   ,SPASM
6E 2160 DATAWEAK,POORLY,*
40 2170 DATANOISE,DIN(RACKET),R
   OW,LOUD,BELLOW,SHOUT,GROWL,H
   OWL,*
5F 2180 DATAHOT,WARM,SHIMMER,SI
   MMER,BOIL,SCORCH
81 2190 DATASEARING,BLISTERING,
   TORRID,HAZY,*
D4 2200 DATACOLD,FROZEN,FREEZIN
   G,CHILLED,ICY(COLD),*
FB 2210 DATAQUIET,SOFTLY,GENTLY
   (SOUND),WHISPER,HUSH,SUBDUED
   ,*
75 2220 DATAHATE,ABHORE,DISLIKE
   ,SHUN,*
96 2230 DATAABILITY,SKILL,DEXTE
   RITY,PROWESS,TALENT,CLEVER
45 2240 DATAEXCEPTIONAL,*
AF 2250 DATAEFFORTLESS,EXPRESS,
   PROPULSIVE
76 2260 DATASUDDEN,ABRUPT,QUICK
   ,HASTY,FAST,SPEEDY,BRIEF(SPE
   ED),*
CD 2270 DATASLOW,SLOVENLY,DULL,
   *
72 2280 DATAMACHINE,GADGET,EQUI
   PMENT,*
57 2290 DATADEVICE,IMPLIMENT,TH
   ING,ARTICLE,OBJECT,*
6A 2300 DATAWET,DAMP,DRIPPING,S
   ODDEN,SOGGY,SOAKING,MOISTE,L
   IQUID,WATER,*
DC 2310 DATAHOUSE,SHOP,SCHOOL,F
   LAT,BUNGALOW,OFFICE,HUT,PRE-
   FAB,TOWER,COTTAGE
EF 2320 DATAWORKS,FACTORY,WORKS
   HOP,STORES,STATION,AIRPORT,T
   HEATRE,UNIVERSITY
F8 2330 DATACOLLEGE,POLYTECHNIC
   ,HOSPITAL,CONDOMINIUM,CHURCH
   (STRUC)
12 2340 DATABUILDING,STRUCTURE(
   BLDING),HOVEL,FOLLY
3C 2350 DATAEDIFACE,FACADE(BUIL
   DING)
12 2360 DATADESIGN,ARCADE,PREMI
   SES,*
8B 2370 DATAEDIT,JUDGE,SCRUTINI
   ZE,CHECK(SEE),OVERSEE,INVEST
   IGATE,CRITICIZE,OBSERVE
A7 2380 DATAVERIFY,*
4D 2390 DATATERRACE,AVENUE,CRES
   CENT,TOWERS,MEWS,CLOSE,CUL D
   E SAC,SQUARE(ADDRESS)
34 2400 DATABOULEVARD,*
29 2410 DATAFIND,DISCOVER,COME
   ACROSS,FIND OUT,EXPLORE,*
11 2420 DATAINVENT,FORMULATE,CO
   NCOCT(CREATE),DEVISE,MAKE UP
   ,PLOT,CREATE,*
A0 2430 DATASTORM,TEMPEST,TURBU
   LENCE,*
01 2440 DATAPUNCTURE,IMPACT,IMP
   ALE,PIERCE,HOOK,*
85 2450 DATAINFLUENCE(HINT),SUG
   GEST(IDEA),IMPLY,PROPOSE,HIN
   T,INSINUATE,INFER
DB 2460 DATACONVEY(SUGGEST),TEM
   PT,*
71 2470 DATADROP,PLUMMET,DIVE,S
   LIP(FALL),FALL,SINK(FALL),*
15 2480 DATAFLY,CLIMB,RISE,ASCE
   ND,*
46 2490 DATATHROW,CHUCK,SLING,C
   AST,FLING,PROJECT,FIRE(GUN),
   PROPEL,*
F6 2500 DATADISASTER,DEVASTATIO
   N,CARNAGE,DESTRUCTION
3D 2510 DATABROKEN,DAMAGE,WRECK
   ED,BUST,*
21 2520 DATAENLARGE,INCREASE,MA
   GNIFY,MAKE BIGGER,AUGMENT,WI
   DEN,PROPAGATE,GROW
6F 2530 DATAESCALATE,EXPAND,INT
   ENSIFY
91 2540 DATAMULTIPLY,FORTIFY,*
A5 2550 DATAROUND,CIRCLE,CIRCUL
   AR,SPHERE,CIRCUIT,ENCLOSURE
6C 2560 DATAREVOLVE(SHAPE),ELIP
   SE,*
79 2570 DATAARMY,PLATOON,SQUADR
   ON,REGIMENT,DETATCHMENT,TROO
   PS,CORPS,FORCE
E4 2580 DATADIVISION,COMPANY,CA
   VALRY,SOLDIER,COMBATANT,*
5A 2590 DATAMETHOD,SYSTEM,TECHN
   IQUE,STYLE,*
01 2600 DATABASE,LOWEST,BOTTOM,
   DEEPEST,*
39 2610 DATAFIRE(APPLIANCE),RAD
   IATOR,HEATER,RADIANT,OVEN,CO
   NVECTOR,BOILER,*
22 2620 DATADIG,GOUGE,TEAR,SCAR
   (TEAR),RIP,PLOUGH,*
B1 2630 DATAHAPPY,LAUGHTER,SMIL
   E,JOY,EXCITED,SATISFIED,*
10 2640 DATATAP(WATER),VALVE,FA
   UCETT,*
F9 2650 DATATAP(HIT),STRIKE,HIT
   ,THUMP,RAP,WALLOP,BLOW,*
26 2660 DATATANGLE,INTERTWINE,I
   NTERMINGLE,MINGLE,CONFUSE,EN
   TRAP,*
B3 2670 DATAINVERT,CAPSIZE,UPSI
   DEDOWN,TURNOVER,*
9E 2680 DATACAPTURE,ATTAIN,AQUI
   RE(THEFT),STEAL,*
02 2690 DATAHOLD(GRIP),GRASP,GR
   IP,RESTRAIN(GRIP),WITHOLD,*
FE 2700 DATADATA,INFORMATION,IN
   STRUCTIONS,DIRECTIONS,*
28 2710 DATACUP,MUG,BEAKER,CHAL
   ICE,GRAIL,*
57 2720 DATABIRD,HAWK,EAGLE,BUZ
   ZARD,KESTREL,FINCH,TIT,VULTU
   RE,*
9F 2730 DATAPIPE,TUBE,CAPILLARY
   ,VEIN,ARTERY,DRAIN,*
D6 2740 DATAIMMORTAL,DIVINE,UNF
   ADING,FOREVER,*
86 2750 DATAIMPASSIVE,UNEMOTION
   AL,SERENE,UNFEELING,*
F5 2760 DATAPLUNGE,IMMERSE,SINK
   (BURY),EMBED,BURY,*
6D 2770 DATASCAREY,FRIGHTENING,
   TERRIFYING,MONSTEROUS(DEADLY
   ),WORRYING,EVIL,DEADLY
BF 2780 DATADANGEROUS,*
16 2790 DATAUGLY,MONSTEROUS(UGL
   Y),HIDEOUS,VILE,GARGOYLE,*
95 2800 DATABEAUTIFUL,GORGEOUS,
   FABULOUS,PRETTY,HANDSOME,DEL
   ECTABLE,BONNY,RAVESHING
F8 2810 DATAPULCHRITUDE,*
FE 2820 DATASKINNY,THIN,EMACEAT
   ED,STARVED,UNDERNOURISHED,*
29 2830 DATAFAT,PLUMP,OBESE,COR
   PULENT,PORTLY,*
06 2840 DATABOOK,VOLUME,PUBLICA
   TION,ANNUAL,PERIODICAL,ESSAY
   ,FOLIO,THESIS
B0 2850 DATAPAPER,PARCHMENT,MAN
   USCRIPT,SHEET,DOCUMENT
FE 2860 DATABRIEF(LITT),DIRECTO
   RY,MAGAZINE,SCRIPT,*
E8 2870 DATACONTEST,COMPETITION
   ,TOURNAMENT,TEST,QUIZ,HEAT(C
```

```
    ONTEST)
B0  2880 DATACOMPETE,TRY(RACE),R
    ACE(COMPETITION),CHALLENGE,F
    IGHT,RIVAL,*
B4  2890 DATATRIUMPH,WIN,DEFEAT,
    ROUT.*
0D  2900 DATASTAR,MOON,PLANET,UN
    IVERSE,GALAXY,SOLAR SYSTEM,M
    ILKY WAY,RED GIANT
81  2910 DATABLACK HOLE,WHITE DW
    ARF.METEOR,COMET,SHOOTING ST
    AR
61  2920 DATAASTERIOD,*
86  2930 DATADELUDE,FOOL,DECEIVE
    ,DUPE,FALSIFY.DEFRAUD,LIE,FI
    B.UNTRUE,DECEPTION
74  2940 DATACONCOCT(LIE),*
CF  2950 DATAFANCY,ORNATE,COMPLI
    CATED,*
5A  2960 DATAINCREDIBLE,UNBELIEV
    ABLE,AMAZING,FANTASTIC(FANTA
    SY)
36  2970 DATAFALLACY,INCONCEIVEA
    BLE,ASTOUNDING,UNTRUTHFUL,*
F3  2980 DATAFANATIC,ENTHUSIAST,
    FOLLOWER,DISCIPLE,PROTEGE
C4  2990 DATALEARNER,PUPIL,SCHOL
    AR,BEGINNER,APPRENTICE,TRAIN
    EE,NOVICE,*
FD  3000 DATAPERSONIFY,IMPERSONA
    TE,CARACTATURE,*
E6  3010 DATAWRONG,INAPPROPRIATE
    ,INCORRECT,MISTAKEN,*
C6  3020 DATAYOUNG,IMMATURE,CHIL
    DISH,JUVENILE,*
9D  3030 DATARIOT.DISTURBANCE,PU
    BLIC NUISANCE,*
30  3040 DATACONTROL,INFLUENCE(C
    ONTROL),ORDER(COMMAND)
8F  3050 DATACOMMAND,DIRECT,LEAD
    ,SUMMON(COMMAND)
19  3060 DATAPRESIDE,ADMINISTER,
    CONDUCT,MANAGE,OFFICIATE
A5  3070 DATARUN(CONTROL),SUPERV
    ISE,*
65  3080 DATAREMOTE,FAR AWAY,ISO
    LATED,SECLUDED,PRIVATE,*
FC  3090 DATASTICK,FIX(STICK),GL
    UE,ADHERE,CEMENT,BOND,TIE,HO
    LD(TIE),*
D8  3100 DATATRAITOR,DISLOYAL,DI
    SHONEST,ENEMY,*
D0  3110 DATATRAMP,VAGRANT,*
E1  3120 DATADISGUST,AVERSION,IN
    DIGNATION,REPUGNANCE,*
FB  3130 DATAEASY,SIMPLE,*
52  3140 DATATRANSPORT,CARRY,CON
    VEY(TRANS),FORWARD,MAIL,*
CE  3150 DATATRANSMIT,BROADCAST,
    TELL,ADMIT,CONVEY(TELL),SPEA
    K,TALK,*
11  3160 DATAACT,PORTRAY,RECREAT
    E,PRETEND,*
3D  3170 DATAREPAIR,FIX(MEND),MA
    INTAIN,MEND,*
9F  3180 DATAIMPLORE,BEG,ASK,PER
    SUADE,SUGGEST(ASK),SOLICIT,B
    ESEECH,*
27  3190 DATAORDER(FORM),FORM,SH
    APE,ORGANISATION,STRUCTURE(F
    ORM),LAYOUT,*
F8  3200 DATATRY(STRIVE),STRIVE,
    ATTEMPT,MAKE AN EFFORT,DO,*
E7  3210 DATAFASTIDIOUS,FUSSY,SE
    LECTIVE,CHOOSEY,FINICKY,*
54  3220 DATASLIPPY,SMOOTH,ICY(S
    LIPPY),FRICTIONLESS,*
82  3230 DATASLIP(SLIDE),SLIDE,*
53  3240 DATATHOUGHTLESS,TACTLES
    S,CARELESS,CAREFREE,*
6F  3250 DATATACTICS,PLAN,PLANS,
    STRATEGY,IDEA,*
2D  3260 DATACHANGE,ALTER,RECONF
    IGURE,REALIGN,RESET,*
E2  3270 DATAGENTLE(KIND),KIND,B
    ENIGN,GRACIOUS,SALUTARY
EE  3280 DATATENDER,GOOD WILL,TH
    OUGHTFUL,CARING
9F  3290 DATAMILD,SOOTHING,CALM,
    BENEFICIAL,BENEVOLANT,*
F9  3300 DATASTRUGGLE,FLOUNDER,G
    IVE IN,GIVE UP,*
4B  3310 DATAIMPRESS,IMPRINT,STA
    MP,ENGRAVE,MARK,*
B2  3320 DATACRESCENT(SHAPE),ARC
    ,SICKLE,CURVE,*
00  3330 DATAREVOLVE(ROTATE),TUR
    N AROUND,ROTATE,SPIN,*
2C  3340 DATABRISTLE,HAIR,PRICKL
    E,SPINE,STUBBLE,THORN,WHISKE
    R,NEEDLE,*
E8  3350 DATASOIL(AGRIG),CLAY,DI
    RT,DUST,EARTH,GROUND,LOAM,SA
    ND,*
5F  3360 DATASOILED(DIRT),BEDRAG
    GLED,FOULED,GRIMMY,DIRTY
18  3370 DATAMUDDY,POLLUTED,SMEA
    RED,SPATTERED,SPOTTED
CF  3380 DATASTAIN,TARNISH,FILTH
    ,*
C9  3390 DATAWRECKAGE,DEBRIS,FRA
    GMENTS,HULK,REMAINS(WRECK),R
    UBBLE,RUIN,WRACK,*
4A  3400 DATACREST,APEX,PEAK,PIN
    NACLE(SHAPE),SUMMIT,TOP,*
11  3410 DATAREVOLVER,GUN,FIREAR
    M,MACHINE GUN,CANNON,CHAIN G
    UN,BAZOOKA,CARBINE
60  3420 DATAMAGNUM(GUN),*
71  3430 DATASHIP(WAR),FRIGGATE,
    CRUISER,BATTLESHIP,CORVETTE
13  3440 DATADESTROYER(SHIP),TOR
    PEDO-BOAT,DREADNOUGHT,AIRCRA
    FT CARRIER,*
4E  3450 DATASHIP(MERCHANT),STEA
    MER,TUG,KETCH,BRIG,TEA CLIPP
    ER,BOAT,CRAFT
2F  3460 DATAYACHT,PACKET(SHIP),
    LINER(SHIP),FREIGHTER
E6  3470 DATAFREIGHTER,VESSEL(SH
    IP),TRAWLER,WHALER,LAUNCH(SH
    IP),SPEEDBOAT
94  3480 DATACANOE,FERRY(SHIP),D
    REDGER,*
96  3490 DATASOBRIETY,TEMPERENCE
    ,RESTRAINT(ABSTAIN),MODERATI
    ON,ABSTINENCE,*
D7  3500 DATASNOBBERY,ARROGANCE,
    CONDESCENSION,PRIDE,SNOOTY,*
9F  3510 DATASOCIETY,CULTURE,HUM
    ANITY,MANKIND,POPULATION,SOC
    IAL ORDER,*
9A  3520 DATABRIGHT,BEAMING,BRIL
    LIANT,DAZZLING,GLEAMING,GLIT
    TERING,GLOWING(LIGHT)
EB  3530 DATALUMINOUS,ILLUMINATE
    D,RESPLENDENT,SHIMMERING,SHI
    NING,LIGHT,*
D2  3540 DATAASSUME,PRESUPPOSE,P
    REJUDGE,*
82  3550 DATASTOP,PREVENT,CHECK(
    STOP),HAMPER(STOP),HOLD BACK
    ,AVERT,FOIL(STOP)
88  3560 DATAHINDER,IMPEDE,OBSTR
    UCT,BAR(STOP),*
CF  3570 DATAHAMPER(FOOD),PARCEL
    ,PACKAGE,LETTER,ENVELOPE(POS
    TAL),*
B1  3580 DATATHEOREM,THEORY,DEDU
    CTION,GUESS,FORMULA,HYPOTHES
    IS,PRINCIPLE
C6  3590 DATAINFERENCE,CONCLUSIO
    N,JUDGEMENT,REASON
B4  3600 DATAPROPOSITION,RULE,ST
    ATEMENT,*
A7  3610 DATATHERAPY,CURE,HEALIN
    G,TREATMENT,*
96  3620 DATATHEOLOGY,RELIGION,D
    IVINITY,ECCLESIASTICAL,CHURC
    H(RELIG),DOCTRIN,*
9C  3630 DATATHIEF,BURGLAR,MUGGE
    R,CRIMINAL,EMBEZZLER,LARCENI
    ST,PILFERER,ROBBER
79  3640 DATASWINDLER,SHARK(MONE
    Y),BANDIT,HIGHWAYMAN,CROOK,S
    HOPLIFTER,STEAL
29  3650 DATACHEAT,PICKPOCKET,*
74  3660 DATATHAW,DEFROST,MELT,W
    ARM UP,*
7C  3670 DATATENSION,STRAIN,STRE
    TCH,TAUT,TIGHT,PULL,*
47  3680 DATAWRITING,CALLIGRAPHY
    ,PENMANSHIP,HANDWRITING,SCRA
    WL,SCRIBBLE,*
1D  3690 DATAWRITER,AUTHOR,COLUM
    NIST,ESSAYIST,NOVELIST,WORDS
    MITH,*
60  3700 DATAENVELOP,BLANKET,CLO
    AK,CONCEAL,EMBRACE,ENCASE,EN
    CIRCLE,ENCLOSE
79  3710 DATAENCOMPASS,ENGULF,EN
    WRAP,SHROUD,CIRCUMNAVIGATE,*
4A  3720 DATAWRAPPING,ENVELOPE,J
    ACKET,COVERING(ENVEL),AWNING
    ,SHEATH,SLEEVE,*
AD  3730 DATAENVIRONMENT,ATMOSPH
    ERE(SOCIAL),BACKGROUND,CONDI
    TIONS,CONTEXT
4F  3740 DATADOMAIN,HABITAT,LOCA
    LE,SETTING,SITUATION,SURROUN
    DINGS,TERRITORY,*
C6  3750 DATAENVISAGE,CONCEIVE O
    F,CONCEPTUALIZE,CONTEMPLATE,
    IMAGINE,PICTURE(IDEA)
94  3760 DATATHINK UP,VISUALIZE,
    *
2F  3770 DATACOMRADE,COMPATRIOT,
    COLLEGUE,FRIEND,COUNTRYMMAN,
    CITIZEN,*
18  3780 DATAMENSURATION,ASSESSM
    ENT,CALCULATION,CALIBRATION,
    COMPUTATION
8A  3790 DATAESTIMATION,MEASUREM
    ENT,SURVEY,*
10  3800 DATALEARNED,ACCADEMIC,C
    ULTURED,EXPERT,INTELLECTUAL,
    LETTERED,VERSED
1C  3810 DATAWELL-READ,*
19  3820 DATAGRAVEYARD,CEMETERY,
    BURIAL GROUND,CHURCH YARD,NE
    CROPOLIS,*
8E  3830 DATACONVOKE,ASSEMBLE,CA
    LL TOGETHER,COLLECT
2C  3840 DATACONVENE,GATHER,MUST
    ER,SUMMON(GATHER),*
43  3850 DATACORRESPONDENT,CONTR
    IBUTOR,REPORTER,REPRESENTATI
    VE,*
EA  3860 DATACOUNTLESS,ENDLESS,I
    MMEASURABLE,INCALCULABLE,INF
    INITE,INNUMERABLE
DE  3870 DATALIMITLESS,*
BA  3880 DATACOSMETIC,SUPERFICIA
    L,ON THE SURFACE,GLOSSED OVE
    R,*
4F  3890 DATACOUNTERBALANCE,COUN
```

continued on page 59

Battletech

Giant fighting machines wielding lasers, machine guns and missiles clash in this game of futurisitc combat. It's a game that's remarkable in two respects – not only is it based on a cult series of board games, but also represents Infocom's entry into the world of role-playing games.

In *Battletech*, you are Jason Youngblood, trainee mech pilot and son of the legendary Jeremiah Youngblood. As the game begins, you're a long way from the heat of the battle, in fact you've only just started your training in mech control and weaponry. It's a process that's long and dull, but necessary if you're going to survive the rest of the game. The problem lies with the Kuritans, a nasty bunch who may at any moment invade your base and destroy your entire city.

You always fear such an attack, and expect it any moment (the game can't be this dull for much longer), and sure enough it comes just before you've crammed in all the training you require. Suddenly, you're very alone and unarmed (unless you've had the sense to get yourself a rifle), you set off for Starport to seek out fellow rebels.

Starport is another town that's been seized from the Lyran Commonwealth, and is full of people who'd like to have a shot at you, particularly if they find out that you're a rebel. So after a change of clothes, you try to infiltrate a victory celebration hoping that other rebels will do the same. You're in luck, as you meet your father's old friend Rex, who even has a 'mech' for you, together with a quest. You must gather together all remaining Lyran rebels, find the secret store of mech parts and get off the planet.

To add to your problems, Kuritans will pose as rebels to damage your mission, which will take you into adventure-style situations and battles with mechs and foot soldiers. Combat involving mechs – whether they are the fast but lightly armed locusts, or the juggernaut chameleons – is a dangerous art, in which you must use the terrain and your mech's abilities to outsmart and out-gun your opponent.

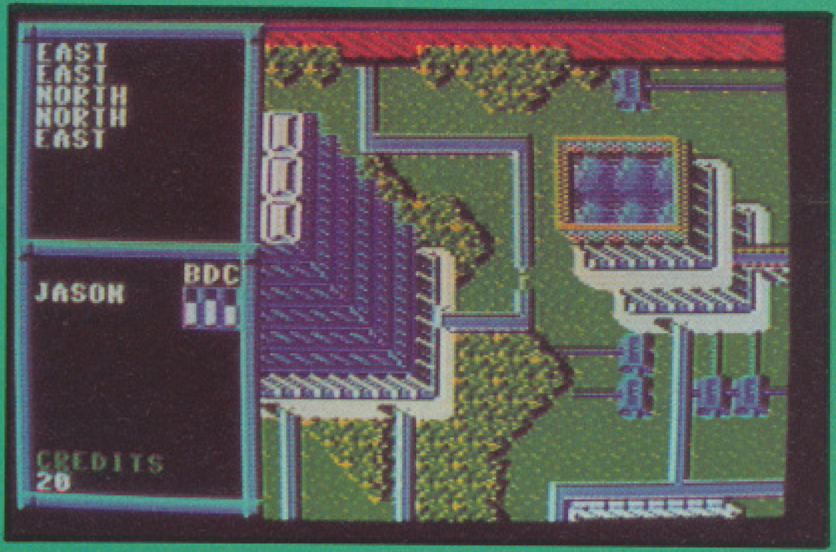
In the game, you must specify a target for each of your weapon systems (a lengthy task for heavily armed mechs), and then plot where your mech will move. Movement and combat are then carried out simultaneously, with text and graphic sequences providing you with a blow-by-blow account. This may take a little getting used to, but it does provide you with the chance to decide your own battle tactics. For example, if you're up against a Locust armed with machine guns, then you'll want to keep your distance and out of their range. Alternatively, a Wasp hurling missiles at you has to be stopped fast, so you want to get as close as possible and add kicking to your arsenal of weaponry.

A mech's main problem is overheating, which can cause a shutdown at the worst possible time. You can delay this by standing in water, but the best way is to conserve your weapons until you really need them. This can get critical if some of your heatsinks get shot up, and the repair bills can be extortionate.

You can increase your chances of success through skills gained either at the training centre or added since, through courses or people who join your party. The most useful are medical skills to patch up your wounded (if a key character such as Jason dies, then your game is over – *a bit like Neighbours, really – Ed*) and engineering, that will allow you to salvage mech parts and ammo from the debris of battle to patch up your own mechs or even build new ones.

Battletech is a massive game, with over 4 million locations to explore, hundreds of mechs to battle, people to deal with and fight, and adventure-style sequences to spice up the action just when you thought it was getting predictable.

Touchline:
Title: *Battletech.* ***Supplier:*** *Infocom (Activision), Blake House, Manor Farm Road, Reading, BERKS., RG2 0JN.* ***Tel:*** *0734-311666.* ***Price:*** *£19.99.*





MacPics on Amiga

How to port those superb black-and-white Mac graphics over to your Amiga

By Jay Gross

Have you ever seen a MacPicture before, one that originates on one an Apple MacIntosh computer? You have to give them credit, their black-and-white graphics really do look nice. But the good news is that you can port those pictures over to the Amiga with a little effort, a little luck, and a kit of utilities from the freely distributable libraries of Amiga software. Oh, and a couple of expensive pieces of commercial software, too, if you want to get really fancy.

To view a MacPicture on the Amiga, you first have to find a way of getting the MacFile into the Amiga. The easiest way is with your old friend, the computer BBS. Many electronic BBSs around the country offer graphic files in the MacPaint picture format. The MacIntosh has many graphics file formats, but MacPaint is one of the most common. Look for MacPaint formatted pictures first, because they're the easiest to view on the Amiga. Download the files from your favourite BBS using your favourite terminal program. If you don't have a modem, you're pretty well stuck for now, as there is currently no way to read MacDisks directly.

If you see files labelled Stuffit, forget them. Stuffit is a MacPacking program like ARC, but is not machine independent, so ask. Maybe the sysoppery where you call will unMacStuff those files for you. While you're downloading, or at your next user-group meeting, try to collect up the following kit of utilities from the public domain libraries:

A: MacView by Scott Evernden – preferably the most recent version.
B: Multiview2.0 by Wayne Hogue. Documentation comes with the first version, and the program isn't terribly easy to figure out without the docs.
C: Your choice of graphic screen saver. My suggestions are Iffencode by Matt Dillon and ScreenX by Steve Tibbet. If you have the commercial program GRABBIT, from Discovery Software, you won't need the others, although ScreenX does some neat additional tricks. In a pinch, you could get by without a screen saver, but it's nicer with than without.

The other pieces of commercial software (besides GRABBIT) are for serious picture porting. First and most important is the $69.95 (ouch! that's the U.S. price) PixMate from Progressive Peripherals. It's by Justin McCormack, and it has a bunch of neat features that will be useful for many other things besides porting MacPictures. If you want to convert larger MacArt to Amiga-sized screens, PixMate is just the ticket. Deluxe Photolab will accomplish the job, but not in the same way as PixMate.

The other item on the shopping list is DigiView 3.0 from NewTek. It's possible that some of the other digitizing software on the market will work, but you'll have to find that out for yourselves. DigiView's software is all you need for this purpose. However, it comes WITH the hardware for $199.95 (U.S. listed price). The current version is 3.0, and it's worth upgrading to if you have the older versions, just to keep the headache quotient down while loading and saving files.

Both Multiview2.0 and MacView will show a MacPicture on the Amiga screen. Multiview offers a number of read options, and will sometimes correctly display MacScreens that MacView cannot cope with. MacView, however, offers the ability to SAVE a MacPaint formatted file, as well as the option to print pictures, and to scroll around the displayed MacPicture in Amiga high-resolution or low-resolution modes.

Try MacView first. If it complains that the file is not MacPaint format,



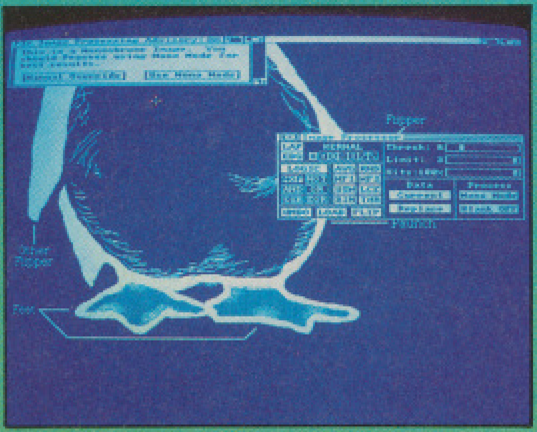
try invoking it from the CLI window with the command:

MacView -f < filename >.

The "-f" in there tells the program not to worry about an incorrect MacPaint file header. The MacIntosh's various softwares aren't too careful about writing that file structure out in an interchangeable manner, so files generated with different products may have variations of the file header.

If the file won't work with MacView, try Multiview2.0. Although Multiview2.0 has an option for saving to IFF file, it won't accomplish what is needed to make PixMate's job fit in only 512K of chip memory. Use

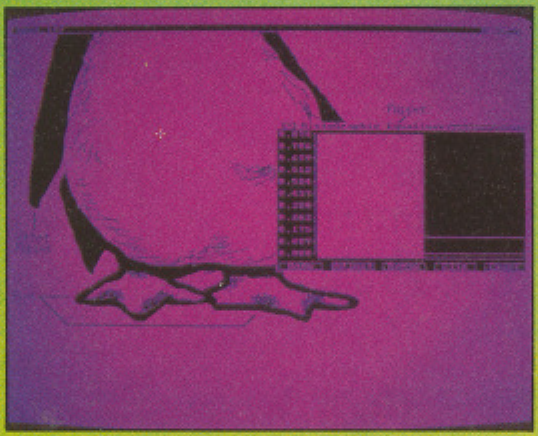
MacView in hi-res. The program has a slider on the right side of the screen for choosing which part of the MacPicture is showing. In addition, you can smooth-scroll the image with the mouse by pointing into the image area, holding the left mouse button down,



and moving the mouse.

Many MacPictures are smaller than the Amiga screen, particularly images intended for use as clipart. If the picture you're trying to import is smaller than the Amiga screen, you're home free. Use MacView's SAVE IFF option, or GRABBIT, or IFFENCODE, or ScreenX to save the screen to an IFF disk file and that's that.

Most MacPictures are, however, more lines tall than MacView (or the Mac) can display at one time. On the original Macs, the screen resolution is 512 x 380, so who but Apple knows why the pictures are bigger than Amiga's 640 x 400 screen? Maybe the purpose is to print a letter-sized sheet and not enjoy the whole thing on the screen at one time at all. But whatever the reason they are bigger.



To get the bigger MacPictures into a single piece viewable on the Amiga, you'll need PixMate's expensive features. If your intent is to use the finished image as line art, you'll also need DigiView 3.0 for the final conversion back to black-and-white. Here's what you do for the bigger ones:

First, with MacView in 640x400 mode, get the top part of the MacPicture to show. Next, save the screen. The latest version of MacView has a nice pulldown menu option for this – appropriately named SAVE IFF. However, if you have an older version, you can multitask some screen saver software to do the job. Using GRABBIT, it's a hotkey operation, with ScreenX, a menu proposition. Using IFFENCODE, you get a CLI, CD to where you want the files to go, as well as IFFENCODE. Left Amiga M shows the picture so IFFENCODE can copy it out to a file. Leave the CLI active and press return when the picture is showing.

The next step is obvious – do the same thing for the bottom half of the MacPicture. Be sure to use a different filename so you'll have two halves in two files. It's best to have some overlap to make lining up the two images easier later on in this involved process. The rest is a job for PixMate.

Before you start, plan on using up some disk space for all this, as the file size for each half just about quadruples. Also it's best to save frequently and with incremental filenames, so if you make a booboo (it's easy to do!), you won't have to start over at Step One. Allow at least 300-K of disk space for EACH 40-50K MacPicture you want to convert. Of course, when the job is done, you can safely get rid of all the extra files.

MacView's display is four colours (note the green gadgets in its title bar). You want greys, not greens, but an IFF screensave will have a four-colour palette to start. You crop out the green gadgets and reformat the picture to allow 16 colours instead of four (memory consumption goes WAY up!). Pictures saved with MacView are two colours, and so are much neater to start with.

Although other products besides PixMate will do the job we've done you need PixMate for the next part: diffusing the black-and-white image into those extra bitplanes, making a picture with shades of grey instead of blobs of black. The purpose of converting to shades of grey is to keep the next step – removal of some of the lines of the image – from making a big mess.

Okay, load the TOP half of your picture into PixMate; use the program's DISPLAY option to turn ON bitplanes 3 and 4. The program will automatically reformat the picture to 16 colours when you exit this option.

To crop off the green gadgets, you can either use the CLEAR AREA command, or scoot the image to the right with SHIFT-<right cursor>. What goes off the right side of the screen is gone forever, so centre the image for further work. Next, do a PACK COLORS and a SORT COLORS (low to hi) on the image, and save the image, in case of disaster. This is the TOP file.

The next step in all this is image processing to "soften" the image into those extra bitplanes. Most of the time, the softening should be done at this point. However, for particularly dense pictures, you might get better (and faster!) results by waiting till after you've reunited the halves to do the softening. Experimentation is the only way to figure out which is which.

Using PixMate's image processing panel, select "AVG." This will take a couple of minutes, but it's very entertaining to watch. The result is a grey-scale picture, and a very good one, too, so, save it just in case.

Next click back to the "DISPLAY" option, and select "COLOR" in the reduction/enlargement side. "Color," rather than "Even," "Odd," or "Average," is usually the best way to make a size change for the MacPictures. With some images, "Average" will work fine, and it's much quicker, so you might want to give it a whirl and see if it works first. Use UNDO if it's not any good.

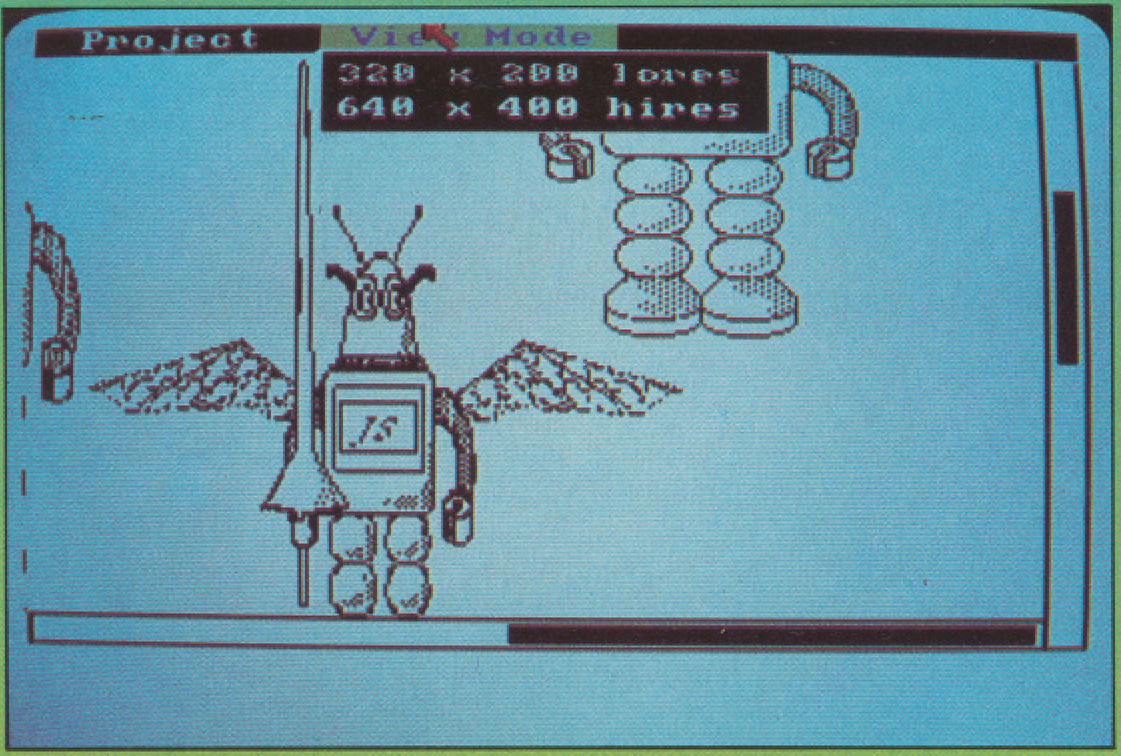
You need to reduce the image size in both directions. The dimension containing the most detail will suffer the worst, so you might need to experiment with choosing "Thinner" or "Shorter" first or second. Use the "Pack Colors" option again, and then do a "Sort Colors." You want "high to low". *Save this image!* It's (finally!) the finished version of the top half of your picture. If you don't like the way it looks, experiment with some of the optional routes described earlier.

Obviously, to get the bottom half, you'll have to go through the same steps. Do the same thing, in the same



order, to get the bottom picture in the same state. You can use the OTHER SCREEN feature to line up the images with each other before you go into image processing, but the AVG function will insist on your closing down the OTHER screen. It's memory intensive, and it wants pure, uncluttered CHIP memory too.

The final step is combining the shrunken, grey-scale images into one. Don't delete these files, however, if you plan to go to the DigiView step – some images work better through that part as halves, rather than wholes. To save memory, you can try reducing the number of colours back to two, after all the diffusing, packing and shrinking has been accomplished. However, you'll lose detail from the picture.



To match the pictures, you want the palettes to be exactly the same, even if you use something besides PixMate to do the joining. To match the pälettes, load both pictures into PixMate at once – one on each screen – pick out the one with the fewest colours, and then invoke PixMate's "Match Palette"/"With Other" option. You should go through this step even if there is no discernible difference between the palettes, and if you plan to work on the images in some other program, you should save the files after the palette matching is complete.

To merge the two images with PixMate, toggle between them with "Flip" and adjust the position of the images with < Shift > < cursor key >. When the two images match up perfectly, use the "Clip" option to get both onto one screen. Save the image, and it's (finally!) done.

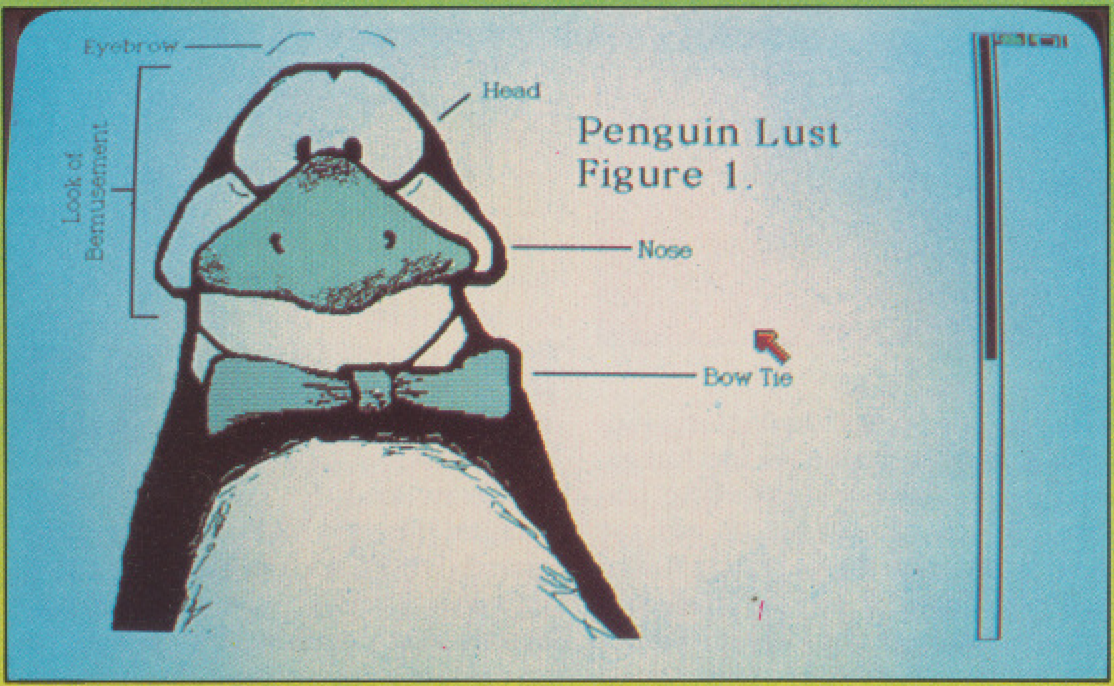
The result so far is a grey-scale picture half the size of the original MacPicture as ported to the Amiga (the MacIntosh uses only a nine-inch screen, so the final image size isn't much different). However, if you want to use the image as line art, say, in a desktop publishing application, you'll want to go the extra mile with DigiView. But take heart – the DigiView part is not nearly as long and involved as the PixMate section.

Also, you can try MacView's IMPORT IFF function. It tries to dither Amiga colour pictures into blobs of black-and-white acceptable to the MacScreen. Sometimes the results are quite nice, but HAM pictures don't make the translation very well. After the MacVersion of the Amiga file is showing, save it either to MacPaint format (for torturing your local MacBBSs) or to Amiga IFF, to get a two-colour image from a colour one. The DigiView step is called for if this doesn't work. Especially if you like playing with DigiView.

First, load up DigiView. You don't need to hook up your camera and lights, because this is a pure software operation. Set DigiView's palette to TWO colours, making the first one black, and the second one white. You'll have to change the second one from grey to white with the sliders. Select FREEZE PALETTE, and click on COLOR. This brings you to the COLOR menu, where you need to reduce the contrast by several clicks (for a very detailed original), and/or raise the sharpness slider by three or four clicks (for a blocky original).

Then, simply LOAD the finished picture that you've saved out of PixMate. DigiView will display its results as it goes, so if you see that the image needs more contrast, brightness, or whatever, stop the process with the left mouse button, make the changes, and select DISPLAY. After some trial and error, you should get a pure black-and-white effect from the finished picture that's every bit as nice as the MacPaint original.

Whew! That's it. The whole process – including the DigiView part – from start to finish. Just think how long would it take you to redraw the picture!



Window on a Maze

Check out this new maze game for younger people that can't cope with the big stuff

By S. T. Burke

Having seen my daughter's face fall enumerable times as an unchivalrous alien brings yet another game to a quick and dissatisfying end (one that everyone else, it seems, can master easily), I decided to have a go at writing a game that would allow her to finish, yet remain a challenge both to her and to those a little more capable.

After much head-scratching, I decided to have a go at that old favourite – the maze, where the object is to find the exit in the shortest possible time. However, my game would have one added challenge – the player would not only decide on the dimensions of the maze, but also the dimensions of the window through which the maze can be viewed. Furthermore, players can decide to create a new maze, or re-run the one before.

Another option is the choice of running speed. Run too fast, and you'll crash into walls. You may also redefine the keys to be used. The game itself, and the 'Maze Generator', are written in Code – I found Basic a little too slow.

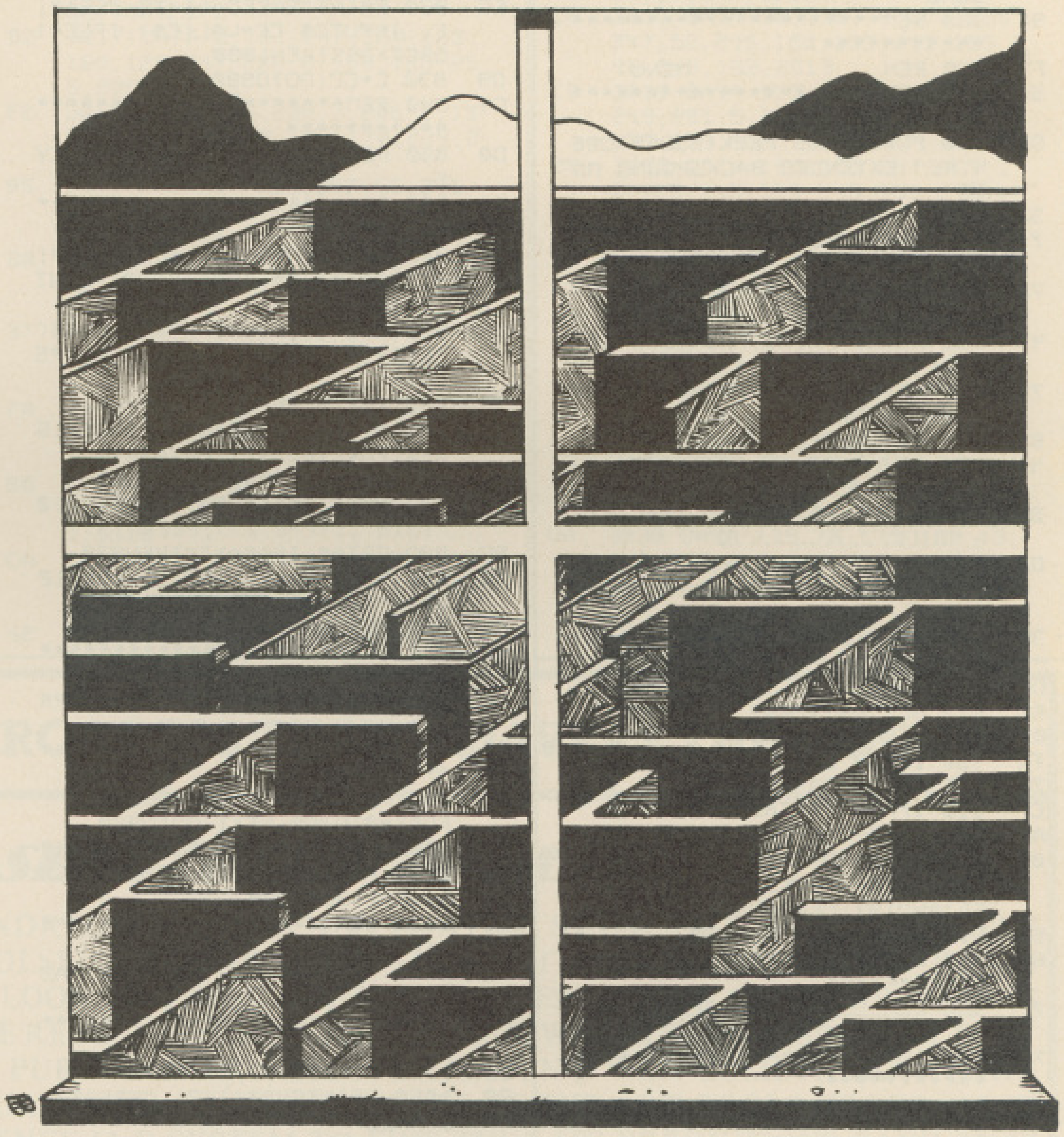
Note that on starting, the following keys are defined:-
Z = left: X = right: RETURN = Up: CSRL/CSRR = Down

Space Bar gives time remaining. (Joystick in port 2 can be used – button = space).

ANY OTHER KEY WILL LEAD TO MENU OR RERUN THE PROGRAM.

Type in and save the listing. The code is included in the basic program. If you wish to increase the maze size, then you can lower variable 'S', which is the approximate end of memory used by the program. One way is to take out the REM statements and make the DATA statements a seperate program. Don't forget to take out the first GOSUB if you do.

Finally, this program makes use of the little-used Extended Background mode of the 64.



WINDOW ON MAZE

```
65   10 REM ***********************
     ***********
BB   20 REM *
             *
7E   30 REM *        WINDOW ON A M
     AZE     *
D7   40 REM *
             *
6D   50 REM *        S.T.BURKE OCT
     .88     *
E3   60 REM *
             *
A9   70 REM ***********************
     ***********
6A   80 :
60   90 :
C2   100 GOSUB2000    : REM POKE M
     /C DATA
54   110 :
FA   120 POKE55,80:POKE56,195:CLR
B8   130 :
C3   140 REM***********************
     ***********
8E   150 REM              FUNCTIONS
EF   160 REM***********************
     ***********
08   170 DEF FN HB(X)=INT(X/256)
E8   180 DEF FN LB(X)=X-FN HB(X)*
     256
31   190 REM***********************
     ***********
C6   200 REM              VARIABLES
5D   210 REM***********************
     ***********
```

```
80  220 S=10000           :REM MEMORY
     START
2A  230 T=40959           :REM MEMORY
     LIMIT
4C  240 L=15:W=15         :REM WINDOW
1B  250 Q=15:K=15         :REM MAZE
0B  260 C=1               :REM SPEED
FE  270 POKE646,1         :REM CURSOR
11  280 POKE53282,2       :REM SURROU
    ND
1B  290 POKE53284,5       :REM WALL
8B  300 POKE53280,0       :REM BORDER
CE  310 POKE53281,2       :REM BACKGR
    OUND
8E  320 REM*********************
    ***********
F5  330 REM               MENU
9A  340 REM*********************
    ***********
9E  350 POKE53265,PEEK(53265)OR6
    4:REM EXTENDED BACKGROUND MO
    DE
32  360 PRINTCHR$(147)
F0  370 PRINTSPC(12);"START.....
    .....S"
7E  380 PRINT:PRINT:PRINTSPC(12)
    ;"ALTER WINDOW...W"
47  390 PRINT:PRINT:PRINTSPC(12)
    ;"ALTER MAZE.....M"
70  400 PRINT:PRINT:PRINTSPC(12)
    ;"ALTER SPEED....*"
B9  410 R$="REDEFINE KEYS"
41  420 PRINT:PRINT:PRINTSPC(12)
    ;R$;"..K"
BF  430 PRINT:PRINT:PRINTSPC(12)
    ;"QUIT...........Q"
DD  440 PRINT:PRINT:PRINTSPC(12)
    ;CHR$(18);"WINDOW ON A MAZE"
    ;
38  450 PRINTCHR$(145);CHR$(145)
    ;
D6  460 POKE53283,2+RND(1)*11
60  470 GET A$:IFA$=""THEN460
0F  480 POKE53283,15
F4  490 IFA$="S"THEN1040
70  500 IFA$= "M"THEN690
CF  510 IFA$= "*"THEN810
AB  520 IFA$= "K"THEN870
04  530 IFA$="Q"THEN POKE55,0:PO
    KE56,160: POKE53265,PEEK(532
    65)AND191:CLR:END
9E  540 IFA$<>"W"THEN460
67  550 REM*********************
    ***********
7F  560 REM              ALTER WINDOW
73  570 REM*********************
    ***********
21  580 PRINT:PRINT:PRINTSPC(8);
63  590 PRINT"LENGTH (ODD 3 TO 2
    5)";Q;
8A  600 INPUTQ$:QQ=VAL(Q$):IFQQ<
    3 OR QQ>25 ORQQ/2=INT(QQ/2)T
    HEN1000
AE  610 PRINT:PRINTSPC(8);
90  620 PRINT"WIDTH  (ODD 3 TO 3
    9)";K;
B5  630 INPUTK$:KK=VAL(K$):IFKK<
    3 OR KK>39ORKK/2=INT(KK/2)TH
    EN1000
DC  640 IF S+(W+KK-1)*(L+QQ-1)>T
     THEN 1000
9B  650 Q=QQ:K=KK:GOTO990
19  660 REM*********************
    ***********
6A  670 REM              ALTER MAZE
25  680 REM*********************
    ***********
9C  690 PRINT:PRINT:PRINTSPC(9);
D8  700 LL=L
55  710 PRINT"LENGTH..........."
    ;L;:INPUTL$:LL=VAL(L$):IFLL<
    1THEN1000
1F  720 PRINT:PRINTSPC(9);
62  730 TT=INT(((T-S)/(LL+Q-1))-
    (K-1))
BF  740 IFTT<1 THEN1000
6E  750 L=LL
7F  760 PRINT"WIDTH 1 TO";TT;"..
    ";W;:INPUTW$:WW=VAL(W$):IF W
    W<1 OR WW>TT THEN1000
69  770 W=WW:GOTO990
80  780 REM*********************
    ***********
89  790 REM            ALTER SPEED
AC  800 REM*********************
    ***********
41  810 PRINT:PRINT:PRINTSPC(10)
    ;
EC  820 PRINT"SPEED (1 TO 255)";
    C;:INPUTC$:CC=VAL(C$):IFCC<1
    ORCC>255THEN1000
D9  830 C=CC:GOTO990
C4  840 REM*********************
    ***********
D0  850 REM            REDEFINE KEY
    S
D0  860 REM*********************
    ***********
73  870 PRINTCHR$(147):PRINT:PRI
    NTSPC(14);R$:FORF=0TO3:PRINT
    :PRINT
01  880 PRINT:PRINTSPC(15);MID$(
    "LEFT.RIGHTUP...DOWN.",1+F*5
    ,5)+".....":NEXTF
54  890 X=1:FORF=0TO3
60  900 X=X*-1:POKE1288+F*160,15
    6+X:POKE1289+F*160,156+X
4D  910 GETA$:IFA$=""THEN900
8E  920 POKE VAL("497"+MID$("010
    71319",1+F*2,2)),PEEK(197)
9C  930 GETA$:IFA$<>""THEN930
A0  940 POKE1288+F*160,15:POKE12
    89+F*160,11
81  950 NEXTF
4C  960 REM*********************
    ***********
F2  970 REM            OK  OR   ERR
    OR!
58  980 REM*********************
    ***********
D9  990 PRINT:PRINTSPC(19);"OK":
    FORF=0TO300:NEXT:GOTO360
0D  1000 PRINT:PRINTSPC(17);"ERR
    OR!":POKE53280,11:FORF=0TO50
    0:NEXT:GOTO300
10  1010 REM********************
    ***********
D3  1020 REM            CREATE MAZ
    E
AF  1030 REM********************
    ***********
D5  1040 POKE53283,9 :REM PATH
20  1050 PRINTCHR$(147);"CREATIN
    G MAZE"
0E  1060 POKE53281,0
1D  1070 B=K-1:A=Q-1
E1  1080 POKE 49680,C
3B  1090 POKE 49198,FN HB(W)
3E  1100 POKE 49186,FN LB(W)
D2  1110 POKE 49204,B
B5  1120 POKE 250,FN HB(S)
EA  1130 POKE249,0:POKE49153,FN
    LB(S)
27  1140 SM=S+B/2+(A/2)*(W+B):RE
    M MAZE START
6F  1150 POKE 252,FN HB(SM):POKE
    251,0
D8  1160 POKE 49170  ,FN LB(SM)
3B  1170 E=S+(W+B)*(L+A):     RE
    M MEMORY END
E4  1180 EM=E-B/2-(A/2)*(W+B):RE
    M MAZE END
D6  1190 POKE 254,FN HB(EM)
53  1200 POKE 253,FN LB(EM)
31  1210 X=INT(RND(1)*L)*(W+B)
C9  1220 Y=INT(RND(1)*W)
6A  1230 P=SM+X+Y            :RE
    M START POS
93  1240 POKE 3,2:POKE4,FN LB(2*
    (B+W))
F9  1250 POKE 3,2:POKE4,FN LB(2*
    (B+W))
55  1260 POKE 5,0:POKE6,FN HB(2*
    (B+W))
E7  1270 POKE182,1:POKE183,FN LB
    (B+W)
7B  1280 POKE184,0:POKE185,FN HB
    (B+W)
10  1290 POKE 248,FN HB(P)
6D  1300 POKE 247,FN LB(P)
08  1310 FORF=0TO3:POKE49920+F,0
    :NEXT
17  1320 :
68  1330 SYS49152:REM CALL MAZE
    ROUTINE
7B  1340 :
EE  1350 REM********************
    ***********
60  1360 REM          PRODUCE WIND
    OW
FA  1370 REM********************
    ***********
2A  1380 Z=-(K-1)/2-(Q-1)/2*(B+W
    ):Z=ABS(Z)
3E  1390 D=P-Z
8E  1400 POKE49921,FN HB(D)
40  1410 POKE49920,FN LB(D)
EB  1420 U=1024+INT(((25-Q)/2)*4
    0-(40+K)/2)
6C  1430 POKE49923,FN HB(U)
67  1440 POKE49922,FN LB(U)
6D  1450 POKE49924,K-1:POKE49928
    ,Q-1
E7  1460 POKE49927,FN HB(Z)
EC  1470 POKE49926,FN LB(Z)
04  1480 POKE 182,FN HB(B+W)
5C  1490 POKE 180,FN LB(B+W)
92  1500 POKE 181,1:POKE183,0
BE  1510 TI$="000000"
C9  1520 PRINTCHR$(19);"[SPC39]"
C5  1530 :
7A  1540 SYS49495:REM CALL WINDO
    W  ROUTINE
2E  1550 :
01  1560 PRINTCHR$(19);"TIME=";T
    I$;
5B  1570 IFPEEK(56320)=111ORPEEK
    (197)=60THEN1590
3C  1580 GOTO1520
00  1590 PRINTCHR$(19);"TIME=";T
    I$
91  1600 IFPEEK(56320)=111ORPEEK
    (197)=60THEN1590
41  1610 GOTO1520
2F  1620 PRINTTAB(19);"ANOTHER G
    O? FIRE-Y/N"
74  1630 IFPEEK(56320)=111THEN16
    70
A9  1640 GETA$:IFA$=""THEN1630
81  1650 IFA$="N"THEN310
10  1660 IFA$<>"Y"THEN1630
61  1670 PRINTCHR$(147)
0E  1680 POKE PEEK(248)*256+PEEK
    (247),160
53  1690 POKE 248,FN HB(P)
5A  1700 POKE 247,FN LB(P)
7A  1710 POKE FN HB(P)*256+FN LB
    (P),170
A4  1720 GOTO1390
72  2000 REM********************
    ***********
33  2001 REM        MACHINE CODE D
    ATA
3C  2002 REM********************
    ***********
42  2003 POKE646,1 :POKE53281,0
    :POKE53280,0:PRINTCHR$(147)
2E  2004 I=49152:L=3000:PRINTSPC
    (5);"POKING MACHINE CODE DAT
    A >"
50  2005 PRINTCHR$(145);SPC(31);
    38-M;CHR$(157);" "
C1  2006 B=0:FORF=0TO14:READA:IF
    A>255THEN2008
72  2007 B=B+A:POKEI+J+F,A:NEXTF
```

```
41  2008 READ C : IF C<>B THEN P
    RINT CHR$(147); "*ERROR IN L
    INE >";L+M : STOP
1A  2009 J=J+15:M=M+1:IF I+J<497
    26THEN2005
6F  2010 RETURN
92  3000 DATA 160,0,169,96,145,2
    49,200,208,251,230,250,165,2
    50,201,160,2734
A9  3001 DATA 208,241,160,50,162
    ,0,134,2,169,224,145,251,32,
    61,192,2031
D3  3002 DATA 232,240,6,224,175,
    240,6,208,240,230,2,208,246,
    165,2,2424
71  3003 DATA 201,0,240,2,208,22
    8,162,14,32,61,192,202,208,2
    50,240,2240
6C  3004 DATA 214,200,208,2,230,
    252,165,252,197,254,208,4,19
    6,253,240,2875
76  3005 DATA 1,96,104,104,162,0
    ,169,4,129,247,32,78,193,134
    ,255,1708
22  3006 DATA 134,180,173,1,195,
    205,3,195,144,30,208,8,173,2
    ,195,1846
0F  3007 DATA 205,0,195,176,20,1
    73,0,195,141,2,195,173,1,195
    ,141,1812
72  3008 DATA 3,195,165,249,133,
    253,165,250,133,254,224,2,14
    4,17,165,2352
AB  3009 DATA 247,24,117,1,133,2
    49,165,248,117,3,133,250,41,
    0,240,1968
86  3010 DATA 13,165,247,56,245,
    3,133,249,165,248,245,5,133,
    250,162,2319
26  3011 DATA 0,161,249,201,224,
    208,68,165,180,129,249,170,2
    24,2,144,2374
29  3012 DATA 17,181,180,24,101,
    247,133,251,165,248,117,182,
    133,252,41,2272
7B  3013 DATA 0,240,17,181,182,1
    33,181,165,247,56,229,181,13
    3,251,165,2361
19  3014 DATA 248,245,184,133,25
    2,169,160,162,0,129,251,165,
    249,133,247,2727
90  3015 DATA 165,250,133,248,23
    8,0,195,208,3,238,1,195,76,8
    5,192,2227
AD  3016 DATA 166,180,232,138,41
    ,3,170,228,255,240,3,76,90,1
    92,162,2176
B2  3017 DATA 0,161,247,168,169,
    160,129,247,206,0,195,169,25
    5,205,0,2311
F2  3018 DATA 195,208,3,206,1,19
    5,192,4,176,40,152,170,224,2
    ,144,1912
85  3019 DATA 17,165,247,56,245,
    1,133,247,165,248,245,3,133,
    248,41,2194
B0  3020 DATA 0,240,28,165,247,2
    4,117,3,133,247,165,248,117,
    5,133,1872
37  3021 DATA 248,41,0,240,11,16
    9,170,162,0,129,247,169,5,12
    9,253,1973
FA  3022 DATA 96,76,85,192,32,15
    8,224,165,140,41,3,170,96,17
    3,0,1651
60  3023 DATA 195,133,2,173,1,19
    5,133,3,173,2,195,133,4,173,
    3,1518
CA  3024 DATA 195,133,5,173,8,19
    5,141,5,195,172,4,195,177,2,
    145,1745
4E  3025 DATA 4,136,16,249,206,5
    ,195,48,30,165,2,24,101,180,
    133,1494
91  3026 DATA 2,165,3,101,182,13
    3,3,165,4,24,105,40,133,4,16
    5,1229
3B  3027 DATA 5,105,0,133,5,41,0
    ,240,211,165,197,201,64,208,
    126,1701
F1  3028 DATA 173,0,220,201,127,
    240,243,201,111,240,114,162,
    0,41,15,2088
74  3029 DATA 74,144,3,232,176,2
    50,138,41,1,240,20,138,74,17
    0,165,1866
0F  3030 DATA 247,24,117,180,133
    ,249,165,248,117,182,133,250
    ,41,0,240,2326
D2  3031 DATA 16,138,74,170,165,
    247,56,245,180,133,249,165,2
    48,245,182,2513
CB  3032 DATA 133,250,162,0,161,
    249,201,5,240,55,201,160,208
    ,33,169,2227
B6  3033 DATA 160,129,247,169,17
    0,129,249,165,249,133,247,16
    5,250,133,248,2843
1A  3034 DATA 165,247,56,237,6,1
    95,141,0,195,165,248,237,7,1
    95,141,2235
55  3035 DATA 1,195,162,20,142,0
    ,196,162,0,202,208,253,206,0
    ,196,1943
9F  3036 DATA 208,246,75,87,193,
    96,162,11,201,12,240,18,162,
    7,201,1920
70  3037 DATA 23,240,12,162,14,2
    01,1,240,6,162,13,201,2,208,
    231,1716
5A  3038 DATA 138,76,175,193,13,
    0,0,0,0,0,0,0,0,0,0, 595
```

Amiga Workbench

More useful hints and tips for Amiga owners from computer buff Burghard-Henry Lehmann

Dear Amiga Fans,
Many people who come to the Amiga from 8-bit machines like the Commodore 64, are in some respects even more confused and daunted by this multitasking, super-graphic, stereophonic sound monster than those for whom the Amiga is their first computer.

I'm talking about the well-known adage about old dogs finding it hard to learn new tricks. After all, those of us – including myself – who grew up on 8-bit machines like the Sinclair Spectrum and the Commodore 64, spend a lifetime (well, about ten years, which in micro computer terms is a lifetime) trying to find out exactly where everything is on the Amiga and how one uses and intercepts Rom routines and so on.

In my time, a mouse was still a pest to poison or shoot or get rid of by some other inhumane method – it was not something that nestled next to your keyboard. So why did I torture myself all those years with rubber keyboards of the Sinclair kind, if it can be that easy?

Now we have to learn about such things as "open architecture", and if you want to use some memory, you have to ask the computer nicely and tell it how much you want and of what type. And then, when you don't need it any more, you must never forget to give it back! But, most confusing of all for the 8-bit brigade, there is no detailed memory map on the Amiga! How can you get used to such a thing?

I'm just joking, because I have to admit that underneath it all, I'm far too much of a computer addict not to be fascinated by it all. I think there are few areas of expertise, short of the philosophical question "Who am I?", that offer more scope for constant expansion of knowledge than computers.

The Amiga is no exception. It's just another stepping stone in the constantly changing sphere of micro electronics and data processing concepts. But let's get back down to earth and try to make sense of some of the new concepts of the Amiga and hope we can understand them. As I've said before, one of the most confusing things about the Amiga is the missing rigid memory map.

But why do we need such a memory map? After all, computer memory consists of nothing but a row of locations, each location given a name in the form of a number. Isn't it much better for the computer itself to take care of the organisation of its memory?

When I started programming on the Amiga, the concept of libraries confused me no end. How can one call a routine in the computer without having anything more than the name of the routine? But let's get this clear first: libraries on the Amiga are basically nothing more than Rom-routines on any other computer. The main difference is that the Amiga has carried the concept of using Rom-routines much further than the older machines.

On most of the 8-bit machines, you only found out how to use the Rom-routines after the machines had been on the market for quite some time, and only because the computer programmers went to the trouble of disassembling the Rom.

On the Amiga, the Rom-routines were from the start structured in such a way that they are accessible to every programmer. Some of the old 8-bit machines were, at least at the beginning, pretty coy about telling people how the operating system worked. The Amiga, on the other hand, was designed to be accessible from the start.

But how can one use something if one doesn't know where it is?

The statement that one doesn't know where each library is is strictly speaking not true and therefore confusing. If you look in the Appendix D-5 of the Amiga Exec manual, you'll find all the library functions of the Amiga listed and in front of each function an address. But this isn't really an address – it's an offset, given in the form of a negative number.

For example, the Exec function AllocMem, which simply enables the programmer to allocate some memory he needs for his program, is located at -198. The base address for all Exec libraries is given in the one fixed memory address in the Amiga: location 4. Get this address and subtract 198 from it, and you've got the calling address of the AllocMem library function.

If you want to call the function of, let's say, the Intuition Library, you have to open that library first. That is, you call the OpenLibrary function of the Exec library, which is at -552, and then Exec gives you the base address for the Intuition Library, that is, where the Intuition Library is located at this moment.

Each function of the Intuition Library has a similar offset. In other words, there is one fixed point, and everything else is related to that fixed point. I sometimes think that the over-emphasis on C on the Amiga doesn't help. On the old 8-bit machines, most people started off with Basic and either stood with it, quite happily, or changed over to assembler.

Basic, of course, keeps you as far away from the real machine as you can get. But, with all its limitations and depending upon the particular Basic dialect the machine allows you to use, at least it keeps things pretty simple. It is really a high level language.

Assembler gets you well and truly into the machine. If you really want to learn about a computer, you should attack it on the machine code level. The main problems with it is that there are a lot of unnecessary myths about machine code programming. Even the Amiga is best explained on machine code level. I just proved this by my explanation of what Amiga libraries are all about.

I could now go on to explain how you pass certain parameters in certain registers before calling a library, and how it hands you certain parameters back in certain registers.

If you look at it from the assembler level, it's all pretty straightforward.

C, on the other hand, is neither a true high-level language, like Basic, which keeps you completely away from the machine, nor to my mind is it as exact in explaining the machine as assembler is.

Somebody said, everybody should learn about computer programming by starting off with assembler. It sounds crazy, but I agree!

Navy Moves

A couple of years ago, a little-known company stunned gamesplayers with a game called *Army Moves*, which was best known for its toughness. It was the game that sorted the men from the boys, as the slightest error resulted in failure. Now here's the sequel – *Navy Moves.*

Your mission is to seek out and destroy an enemy nuclear submarine. It's a task that's made even more difficult by the fact that you're alone. Your mission begins in an inflatable motorboat that's travelling at high speed towards a watery minefield full of floating contact mines that must be hurdled if your mission isn't going to falter before it begins. To add to your problems, enemy troops riding aquatic bikes come hurtling towards you. Luckily, you can fire back to keep them at bay. Soon you'll reach the underwater entrance to the enemy base, which means diving into the depths and tackling scuba divers and sharks.

To get to the base, you'll have to travel through an underwater cavern containing an octopus and a giant sea monster. If you're still hanging on to one of your five lives, you can carry on in part two, which takes you onboard the sub itself.

Now your mission becomes even tougher, as you must plant a bomb at the base of the sub's reactor and then force it to surface so you can transmit a message to your base and get rescued before the bomb explodes. Onboard, you'll have to battle with marines armed with rifles and flamethrowers, and overpower officers to gain their passes (that will give you access to the more secret parts of the subs) and codes (that will allow you to order the sub to surface and transmit the message to your base).

Your realistic chances of completing this mission are slim at best but, above average gamers who like their challenges tough will enjoy *Navy Moves.*

Touchline:
Title: *Navy Moves.* ***Supplier:*** *Langley Business Centre, 11-49 Station Road, Langley, Berks SL3 7YN.* ***Price:*** *£9.95 (cass), £14.95 (disk).*



PC Games

Tony Heatherington assesses the latest games releases on the Commodore PC

688 Attack Sub

688 Attack Sub puts you in control of a top secret billion dollar US submarine in a series of missions that will take you into battle against surface destroyers, helicopters and submarines. These range from training missions against dummy ships to full battle missions in the world's most dangerous sea lanes, and even to the early conflicts of World War III.

To add to the fun, you can also swap to the helm of a Soviet Alfa sub to see how the other half lives, and even play a second player via a modem link which will surely provide the ultimate in submarine simulation.

The sub is controlled through a series of screens manned by your crew, that carry out your orders issued through the mouse or keyboard. For example, at the sonar desk you can deploy and retrieve towed-array sonar, as well as create a 3D-sonar contour map to track enemy vessels. You can also send out active sonar bleeps, but this has its risks as it may alert the enemy to your position.

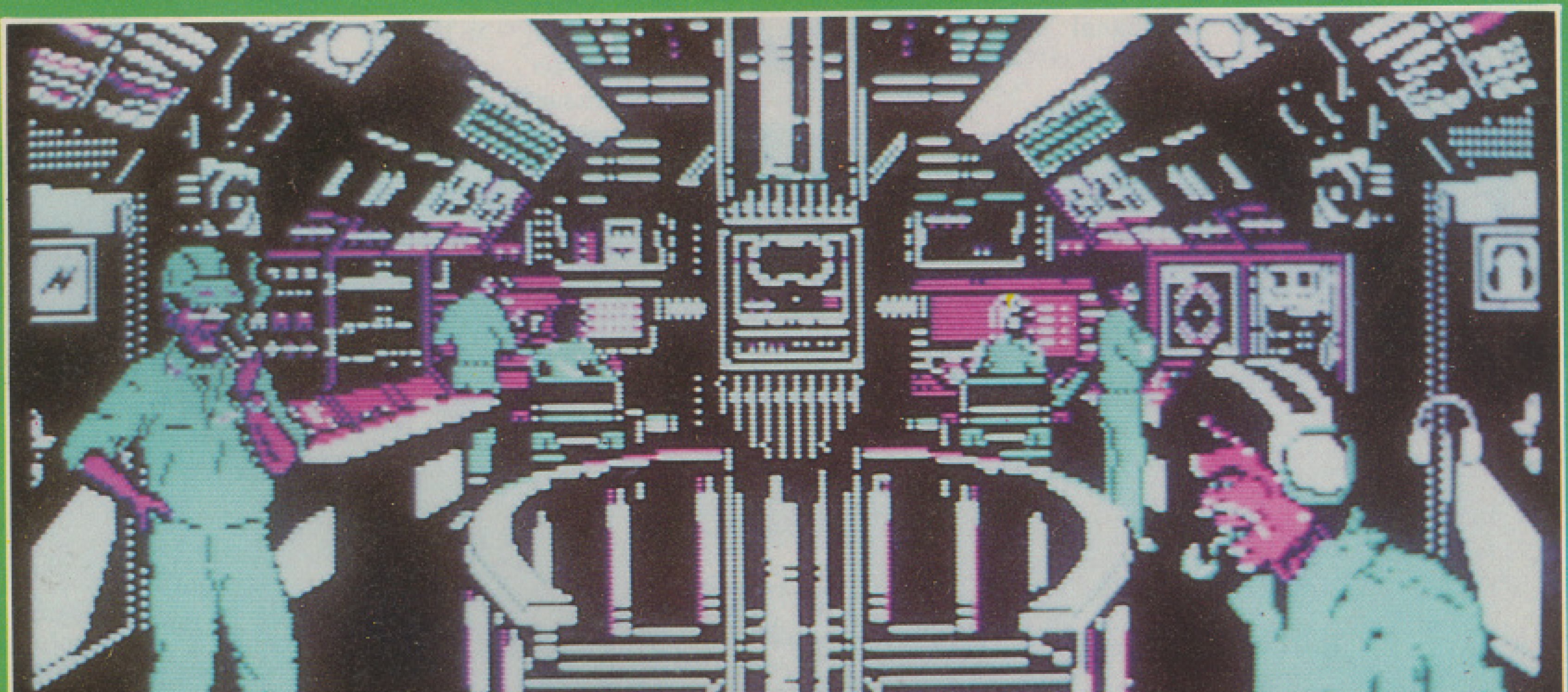
Stealth and silence are essential to submarine warfare, and you must learn how to use the sea's thermal layers to hide behind until you're ready to strike. When you go to battle stations, you have Mark 48 torpedoes and even a few missiles to sink the enemy with, before diving to the depths and safety. Some commanders like to watch the torpedoes strike their targets in glorious 3D, but this can be hazardous, and may even cost you the mission.

If the enemy do track you down, you can either run to safety or fire a noisemaker to decoy their weapons as you escape. The game really comes to life in a full combat situation when you know there's an enemy out there looking for you. If it's a juicy convoy, then you must plan your attack, avoiding the destroyers, subs and helicopters that will be escorting it.

In a one-on-one against an enemy sub, things get really tense as you try and outmanoeuvre your opponent so that you know where he is while remaining hidden from his sonar. However, the sea is a noisy place, so you'll have to develop your own sonar skills to track down your prey.

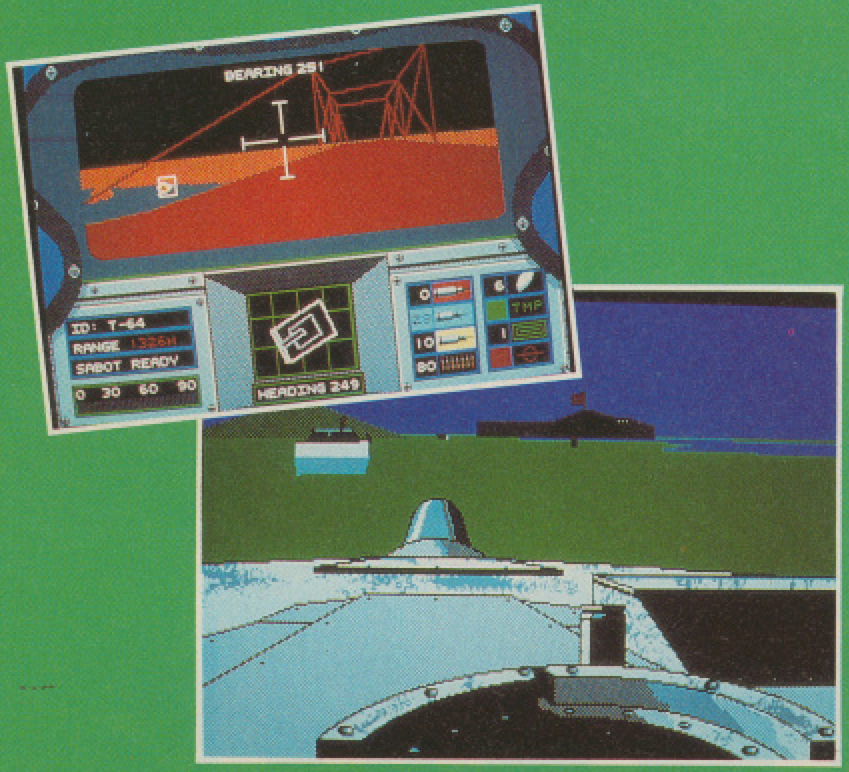
Submarine games are always popular, as they present an irresistible mix of skill and strategy and pile on the atmosphere as you dive, dive, dive into action.

Touchline:
Title: *688 Attack Sub.* ***Supplier:*** *Electronic Arts,* ***Price:*** *£24.99.*



Abrams Battle Tank

Abrams Battle Tank is the land-based equivalent of *688 Attack Sub*, and puts you in control of an M1A1 on the frontline as World War III breaks out. Hundreds of Russian tanks, infantry and helicopters have been detected crossing the border. You're desperately outnumbered and outgunned, but you're determined to go down fighting.



As in *688*, you control the tank from different stations, including those manned by the tank commander, gunner, loader and driver. Your tank is armed with an anti personnel machine gun and a main cannon that can fire either anti-tank or anti-aircraft shells, so it's important to have the right ammo loaded as the threat approaches.

Moving the tank can be a little tricky to start with, as you can either move the tank or the turret through left and right rotate controls and forward and back acceleration.

The gunner's station is probably the one you'll use most as you can drive the tank, load weapons and aim and fire at targets. However you may want to view the driver's station for accurate navigation, and the commander's screens to check fuel and ammo levels, or even go up top to spot approaching aircraft.

When the battle is raging, the battlefield will become strewn with smoking debris which may affect visibility, although you can switch to thermal imaging to continue tracking down your targets. The M1A1 is even fitted with a smoke discharger to hide you from enemies not fitted with thermal imagers.

All these features combine to give you a chance in the battle ahead, but your early attempts may end up in your tank spinning around helplessly as you fire at anything that comes into your sights. However, you will then learn to use your scanner to plot the enemy and begin to use the 3D terrain to your advantage. Also, you'll learn to stay within fuel range of a base where you can rearm and refuel before returning to the battle.

Eight different scenarios tune up your fighting skills before you take on the full World War III campaign.

Touchline:
***Title:** Abrams Battle Tank. **Supplier:** Electronic Arts. **Price:** £24.99.*

Battlehawks 1942

After warfare underwater and battles on land, the fight is taken to the air in this historical simulation of naval air combat in World War II. It's set in the Pacific in 1942, when the USA sprang into action after Pearl Harbour, and clashed with the Japanese navy in four key battles including Midway. However, these naval battles weren't decided by ships' guns but by aircraft that flew dangerous missions from the decks of carriers. In this game you can take on the role of either an American or Japanese pilot.

After a few training missions to spruce up your flying skills, you're ready for action and one of the missions associated with each of the battles. On average there are three missions for each side in each battle, so there's plenty of action waiting for combat pilots. To add to the game, the disks are supported by a 150 page manual packed full of historical mission data and accounts from the real pilots which include useful tips on planning attack runs and ensuring you get back home in one piece.

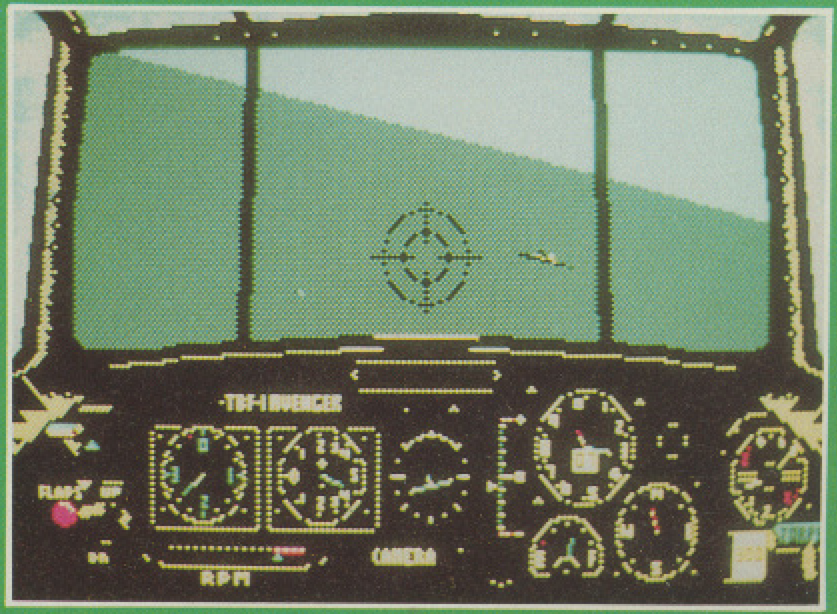
The missions fall into four basic categories – dogfight, escort for bombers, dive bombing and torpedo attacks. The first two are aerial fights high above the waves, where height and speed can give you the edge over your enemy, but it's the bombing and torpedoing missions that will really test your skills.

Your job is to deliver a bomb or torpedo on an enemy carrier that's defended by guns and its own fighters. Sounds impossible, but many have succeeded before you. The idea behind dive bombing is to fly high over the target to avoid contact and then dive down, drop the bomb and climb out of range before you get shot at or engulfed in the explosion.

Torpedo attacks are the other extreme as you avoid detection by flying low – very low – over the sea before launching your torpedo and pulling out to safety. However, airspeed and range are important to ensure that the torpedo hits the target and sends the flattop to the bottom of the sea.

Many of these missions are extremely hazardous, and the aircraft unsuitable to carry them out but success brings medals and promotions and a drive to play the game again and again.

Touchline:
***Title:** Battlehawks 1942 **Supplier:** Lucasfilm Games (US Gold), Units 2/3 Holford Way, Holford, Birmingham.*





Star Trek

Space... The final frontier. This is the game that all trekkies have been waiting for, as you boldly go where no game has gone before. In the game you control Captain James T. Kirk, Spock, Sulu, Chekov, Uhura and Scott in the Enterprise's most challenging mission yet.

The Klingons have developed a ray that causes Federation starships to turn renegade, and this has set the alarm bells ringing in Starfleet Command. Their answer is to set up a guarantee zone around the infected areas and send in the Enterprise to sort out the problem. This works well in theory but the Klingons, Romulans and Federation planets trapped within the Klein Sphere object to this, making your task even more difficult.

The screen consists of a large window surrounded by seven smaller ones that can be selected by clicking on them with a joystick controlled cursor. Through this method you can select the main crew members and control the ship through their departments.

For example, Kirk is in charge of the stores (items that you find on worlds) and decides who forms landing parties by placing them on the transporter. Spock provides information on planets and systems selecting through Sulu's navigation controls. Scott, naturally, controls the warp and impulse engines and warns if the dilithium crystals "canna take it". Uhura sends and receives messages from Starfleet command, while Chekov controls the phasers, photon torpedoes and the combat screens.

The object of the game is to locate the Klingon device and find the means to destroy it. This quest will take you to planets and systems within the guarantee zone and into adventure-like sequences on the surface of life-supporting planets. Whatever the problem, each of your crew will have a different way of approaching it and it's up to you to choose the best way.

For example, a door blocks your path. Sulu and Chekov want to break it down with force, Uhura wants to send signals at it, Scott wants to look for hidden switches and Spock wants to analyse it. Selecting the wrong option may either damage the door or the crew member so you can't just blunder around without applying some logic.

Your reward for a correct solution will be a device that may help solve another problem elsewhere in the zone, or might be part of the final solution.

Whenever an enemy ship approaches the alarm will ring and it's up to you via Chekov to track the enemy ships and blast them with either phasers or photon torpedoes. Any careless shooting could result in damage to the ship, its weapons or the invaluable dilithium crystals. Luckily, the zone is packed full of planets containing refineries to restock crystal supplies and friendly bases for repairs, but also strange worlds that drain your ship's energy. There are 21 different types of worlds in the game, so you never quite know what you'll find in a system.

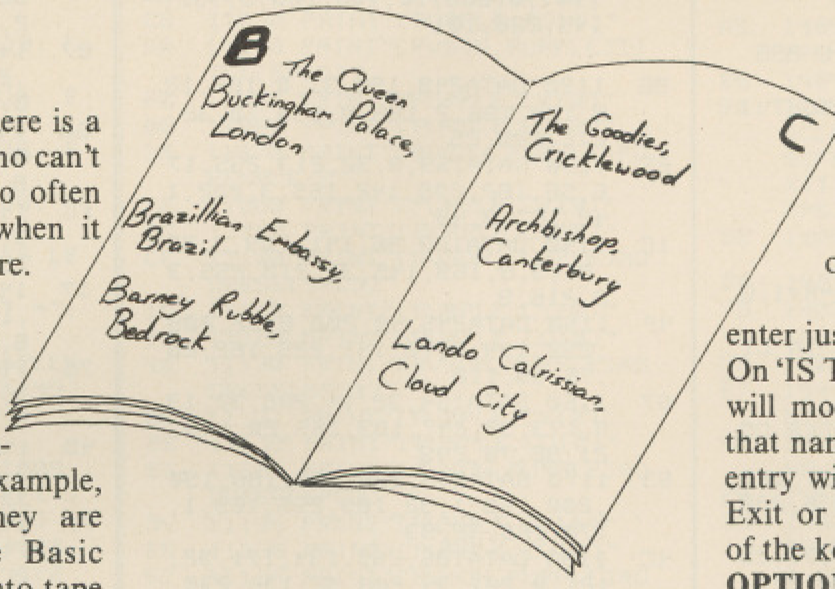
The original ST version of the game finally appeared 18 months late, over a year ago, so it's good to see the C64 version here at last. It contains all the favourite Trek characters (except the security guards with only 10 minutes to live) and is sure to please Commodore trekkies.

Touchline:
Title: *Star Trek.* ***Supplier:*** *Firebird (Microprose), Unit 1, Hampton Street, Tetbury, Gloucs.* ***Tel:*** *0666 54326.* ***Price:*** *£14.99 disk, (£9.99 cass).*

Address Book

Tape users take heart! In this database program, you can tailor for your own needs

By A.E.C. Moore



The program presented here is a boon to all tape users who can't afford a disk drive. Too often we are left out in the cold when it comes to database type software. This program offers a place to keep all those important name and address files, with printout facilities if required. This program has many advantages for the tape user. For example, as the entries are made, they are automatically put into the Basic program itself. When saved onto tape at the finish, the program and its own built-in Turbo loader are saved, thus eliminating the need for a seperate file of addresses, as well as the need to keep track of the cassette counter. You simply have to rewind the tape after loading, and it will save over itself.

Building the program

The program is built by typing in Prog 1, and then saving it. Next, the Turbo part of the program, Prog 2, is typed in and saved. The third program, Prog 3, which is the main body of the program, can now be typed in and saved.

The three programs are now loaded in and run one after the other to make the final version of the Address book program.

A few notes

'Bytes Free' is a true value of the available RAM.

'Entries left' is a guess at the total entries that might go in the program, as the length of each entry is a variable. If at some time this figure is found to be incorrect, the adjustment can be made to the value '52' in line 9070 in the main program.

Any alterations made when the final program is in use should be done after the program has been run, and then stopped in the normal manner. This is because the turbo load part of the program is loaded back with the Basic, giving an incorrect value for the start of Basic variables.

The nature of the program is such that it will always take up the same amount of space on the tape, and therefore will always load back in the same amount of time – 1 min 45 secs. Pressing the 'Left Arrow' key will at most times return you to the menu.

Main program

From the MENU, the following options can be chosen:-

OPTION 1: To enter a name etc. All typed entries are limited to 20 characters. INST/DEL key will delete the last character typed . After each entry, press RETURN. To leave a line blank, just press RETURN.

Surname: Enter Name
Title: Enter Mr A. Mrs S etc. Enter as you require the printout to read. The F1 key will give an auto entry of Mr & Mrs.
Forenames: Enter name/names
Address: Use seperate lines to enter address as required
Tel. No: Enter the number, including spaces is required.
For example – 0442877777 or 0442 877777 or 0442-877777

OPTION 2: To find an entry, enter just the surname.
On 'IS THIS THE RIGHT ONE':- 'N' will move on to the next entry with that name. 'Y' will show the complete entry with options to Change Delete, Exit or Print. Press the capital letter of the key of your choice.

OPTION 3: Will list all entries with options to Change, Delete or Exit.

OPTION 4: Will save the whole program to tape.

OPTION 5: Will tidy up the entries whenever needed. Deletions are not dealt with in the final total until this option is used. Therefore it is suggested that this option is used prior to a re-save.

OPTION 6: Will list through all the entries giving the option to print.

Please note that the printer used was a Citizen 2 colour mini printer, and that using other types of printers may necessitate alterations in the program, namely in line 49660. The size of the label used could also vary with different printers, so some alterations might have to be made, but these should be obvious to you when typing in Prog 3.

Final note

Because of the way the program saves itself out each time, there's no reason why you shouldn't use this program for things other than an Address book. For example, it could be used for Video collections or for the dreaded Train Spotting or... the list is endless. All you have to remember is to keep the fields the same size, and only change the text.

PROG 1

```
73   10 PRINT"[CLR]";
25   20 K=0:H=0:L=0
BF   30 A=000+256*154:C=030+256*1
     54
69   40 FORJ=ATOC
17   50 K=K+1:IFK=17THEN140
13   60 READX
FD   70 L=L+X:IFL>255THENL=L-256
95   80 POKEJ,X
57   90 PRINT"[HOME,DOWN5,RIGHT4]
     "1000+H*10,J,X
30   100 NEXTJ
41   110 READX
7B   120 IFL=XTHEN180
3A   130 GOTO160
63   140 READX
42   150 IFL=XTHENL=0:K=0:H=H+1:G
     OTO50
5B   160 PRINT"[DOWN2]ERROR IN LI
     NE";1000+H*10
2A   170 END
FE   180 PRINT"[DOWN2]TRANSFER CO
     MPLETED":END
A9   1000 DATA169,255,133,251,169
     ,7,133,252,230,251,208,6,230
     ,252,201,150,81
2A   1010 DATA240,11,160,0,177,25
     1,208,240,200,192,3,208,247,
     96,52,237
```

PROG 2

```
73   10 PRINT"[CLR]";
25   20 K=0:H=0:L=0
C4   30 A=090+256*156:C=000+256*1
     60
69   40 FORJ=ATOC
17   50 K=K+1:IFK=17THEN140
13   60 READX
FD   70 L=L+X:IFL>255THENL=L-256
95   80 POKEJ,X
57   90 PRINT"[HOME,DOWN5,RIGHT4]
     "1000+H*10,J,X
30   100 NEXTJ
41   110 READX
7B   120 IFL=XTHEN180
3A   130 GOTO160
63   140 READX
42   150 IFL=XTHENL=0:K=0:H=H+1:G
     OTO50
5B   160 PRINT"[DOWN2]ERROR IN LI
     NE";1000+H*10
2A   170 END
F5   180 PRINT"[DOWN2]TRANSFER CO
     MPLETED":END
D9   1000 DATA56,165,45,133,90,23
     3,130,133,95,165,46,133,91,2
     33,2,133,91
D1   1010 DATA96,165,171,208,6,16
     9,0,162,208,208,4,165,55,166
     ,56,133,180
D6   1020 DATA88,133,174,134,89,1
     34,175,32,191,163,230,89,165
     ,88,166,89,92
81   1030 DATA133,193,134,194,172
     ,0,3,140,166,2,172,1,3,140,1
     67,2,86
F5   1040 DATA141,0,3,142,1,3,164
     ,171,240,4,133,55,134,56,162
     ,255,128
19   1050 DATA32,142,251,32,219,2
     52,32,219,252,32,209,252,176
     ,75,160,0,31
06   1060 DATA177,172,201,190,208
     ,240,232,169,32,224,6,240,13
     ,224,7,208,239
E6   1070 DATA2,169,157,224,11,20
     8,5,169,189,44,169,162,145,1
     72,200,24,2
91   1080 DATA177,172,101,193,145
     ,172,8,200,177,172,201,160,2
     40,249,40,101,204
7B   1090 DATA194,145,172,224,4,1
     76,188,157,170,2,232,136,177
     ,172,157,168,170
D7   1100 DATA2,232,169,76,157,16
     6,2,208,170,96,32,121,0,168,
     32,115,210
3C   1110 DATA0,192,148,208,8,190
     ,104,0,208,10,108,166,2,192,
     149,208,101

96   1120 DATA249,190,31,0,162,12
     8,108,166,2,169,0,133,10,32,
     212,225,25
BF   1130 DATA169,0,32,213,255,17
     6,56,162,209,142,165,3,202,1
     42,167,3,48
1C   1140 DATA169,96,141,209,3,32
     ,81,3,8,169,145,32,210,255,3
     2,216,9
4F   1150 DATA245,40,208,8,32,209
     ,252,144,3,76,141,225,162,28
     ,76,55,112
67   1160 DATA164,32,14,226,32,13
     8,173,32,247,183,165,20,166,
     21,96,76,249
B3   1170 DATA249,224,169,188,190
     ,200,160,1,32,189,255,169,1,
     170,168,32,93
4C   1180 DATA186,255,134,171,32,
     121,0,201,34,208,32,136,230,
     122,177,122,113
89   1190 DATA240,4,201,34,208,8,
     198,171,165,171,208,240,169,
     32,190,199,134
85   1200 DATA1,232,224,17,144,24
     2,32,121,0,240,86,169,34,32,
     255,174,211
9F   1210 DATA240,79,190,87,0,133
     ,78,134,79,190,87,0,133,80,1
     34,81,189
1C   1220 DATA190,10,1,176,230,16
     2,9,190,237,0,157,44,3,202,1
     6,247,82
C9   1230 DATA162,44,160,3,134,19
     3,132,194,162,60,134,174,132
     ,175,70,157,38
06   1240 DATA32,237,245,8,32,21,
     253,6,157,40,176,131,169,16,
     133,183,47
D0   1250 DATA32,147,245,190,19,1
     ,96,46,3,165,1,9,48,133,139,
     208,203
1B   1260 DATA27,160,3,185,43,0,1
     53,78,0,136,16,247,48,183,23
     0,78,51
02   1270 DATA208,2,230,79,165,78
     ,197,80,165,79,229,81,96,88,
     162,0,147
38   1280 DATA160,0,132,192,173,1
     7,208,41,239,141,17,208,165,
     1,133,139,174
6C   1290 DATA41,254,133,1,202,20
     8,253,136,208,250,120,132,83
     ,169,2,190,78
79   1300 DATA138,1,198,83,208,24
     7,169,9,133,83,165,83,190,13
     8,1,198,252
D3   1310 DATA83,208,247,165,78,1
     90,138,1,165,79,190,141,1,16
     5,80,190,73
5E   1320 DATA141,1,165,81,190,14
     1,1,162,4,190,134,1,132,171,
     177,78,233
6C   1330 DATA190,146,1,190,4,1,1
     44,246,234,165,171,190,146,1
     ,190,141,112
41   1340 DATA1,140,160,2,32,147,
     252,165,139,9,48,133,1,133,1
     92,96,114
DA   1350 DATA202,208,253,96,162,
     3,44,162,5,190,134,1,133,189
     ,24,101,115
A3   1360 DATA171,133,171,169,8,1
     33,164,190,190,1,166,163,202
     ,208,253,190,208
BC   1370 DATA190,1,70,189,162,13
     ,144,2,162,41,134,163,198,16
     4,240,17,98
91   1380 DATA162,9,190,134,1,234
     ,240,223,165,1,73,8,133,1,23
     8,32,52
F7   1390 DATA208,96,32,32,32,32,
     32,32,32,32,32,32,32,32,32,3
     2,240
87   1400 DATA32,32,173,17,208,41
     ,239,141,17,208,32,21,253,20
     0,132,192,146
68   1410 DATA169,54,133,1,202,20
     8,253,136,208,250,120,169,24
     8,141,6,221,215
FC   1420 DATA32,228,3,102,189,16
     9,2,197,189,208,245,133,123,
     160,9,32,229
17   1430 DATA212,3,201,2,240,249
     ,196,189,208,230,32,212,3,13
     6,208,246,7
76   1440 DATA133,172,32,212,3,13
     3,173,32,212,3,133,174,32,21
     2,3,133,0
4D   1450 DATA175,132,171,32,212,
     3,145,172,36,16,24,101,171,1
     33,171,32,190
AF   1460 DATA219,252,32,209,252,
     144,236,32,212,3,140,160,2,3
     2,147,252,20
55   1470 DATA165,139,133,1,88,13
     3,192,32,170,245,134,45,132,
     46,165,189,217
4A   1480 DATA197,171,76,154,225,
     169,8,133,163,32,228,3,102,1
     89,198,163,163
14   1490 DATA208,247,165,189,96,
     169,16,44,13,220,240,251,173
     ,13,221,142,103
B2   1500 DATA7,221,72,169,25,141
     ,15,221,104,74,74,96,102,32,
     255,4,76
F3   1510 DATA110,64,239,169,102,
     0,110,255,127,68,102,32,100,
     187,239,78,190
87   1520 DATA255,223,255,171,183
     ,32,110,10,255,0,38,40,231,0
     ,110,64,185
3A   1530 DATA255,4,255,32,255,32
     ,255,64,255,255,100,160,255,
     32,110,215,230
B3   1540 DATA111,255,102,32,102,
     255,255,0,255,76,255,36,255,
     32,33,9,15
BC   1550 DATA239,65,255,0,239,32
     ,255,0,255,0,255,186,255,191
     ,255,223,145
38   1560 DATA255,64,102,32,103,1
     69,238,0,255,68,238,187,119,
     0,255,68,105
FE   1570 DATA255,70,255,169,255,
     33,255,0,255,64,255,32,255,0
     ,255,0,104
BF   1580 DATA254,200,83,68,49,83
     ,148,117
```

PROG 3

```
2C   4 POKE44,8:SYS39424:IFPEEK(2
     51)<253THEN7
1A   5 POKE251,(PEEK(251)-253):PO
     KE252,PEEK(252)+1
7C   6 POKE45,PEEK(251):GOTO8
61   7 POKE45,PEEK(251)+3
```

```
F5  8 POKE46,PEEK(252):POKE56,15
    4:CLR:XX=0
0A  10 PP= 2
62  20 DIMOP$(8,PP)
C5  30 POKE53281,1
3D  40 POKE53280,13
F1  9000 GOSUB38000
E6  9009 PRINT"[CLR,C5,DOWN,SPC1
    0]NAME AND ADDRESS BOOK"
C7  9010 PRINT"[RVSON,CY40,RVSOF
    F]";
02  9020 PRINT"[DOWN2,SPC17]OPTI
    ONS"
0B  9022 PRINT"[SPC17,CT7]"
B6  9030 PRINT"[DOWN,RIGHT4]1. A
    DD AN ENTRY."
05  9040 PRINT"[DOWN,RIGHT4]2. F
    IND AN ENTRY."
F8  9050 PRINT"[DOWN,RIGHT4]3. L
    IST ENTRIES."
8F  9060 PRINT"[DOWN,RIGHT4]4. E
    XIT AND SAVE."
7E  9065 PRINT"[DOWN,RIGHT4]5. F
    ILE MAINTENANCE."
D5  9067 PRINT"[DOWN,RIGHT4]6. P
    RINT ADDRESSES."
B0  9070 PRINT"[DOWN3]ENTRIES LE
    FT ="52-PP;
B5  9075 PRINT" : BYTES LEFT =";
    :SYS39424
D2  9076 PRINT(PEEK(52)*256+PEEK
    (51))-(PEEK(50)*256+PEEK(49)
    )
5F  9080 GETA$
B2  9090 IFA$="1"THENGOSUB11000:
    PRINTBL$;"[BLUE,SPC10]TYPE I
    N THE DETAILS.";:GOTO12000
2F  9100 IFA$="2"THENGOSUB11000:
    PRINTBL$;"[BLUE,SPC10]ENTER
    THE SURNAME.";:GOTO14000
AE  9110 IFA$="3"THEN9400
3B  9120 IFA$="4"THEN9500
A4  9125 IFA$="6"THEN9200
0A  9130 IFA$<>"5"THEN9080
3B  9140 PRINT"[CLR]38000 KB=1":
    PRINT"RUN":POKE631,19:POKE63
    2,13:POKE633,13:POKE198,3:EN
    D
05  9200 PRINT"[CLR,C5,DOWN9,RIG
    HT6]IS THEN PRINTER ON LINE?
      Y/N."
94  9210 GETA$:IFA$=""THEN9210
43  9230 IFA$="N"THENRUN
08  9240 IFA$<>"Y"THEN9210
F8  9245 IFQW=1THENRETURN
8E  9250 GOTO49400
F5  9400 FF=0:GOSUB11000
B7  9410 FF=FF+1
05  9420 IFFF=PPTHENPRINT"[CLR,D
    OWN,RED,SPC14]END OF BOOK":F
    ORDL=1TO3000:NEXT:RUN
9B  9430 IFOP$(1,FF)=""THEN9410
BD  9440 GOSUB17000
D7  9450 FORI=1TONF:PRINTD$(I);X
    $;OP$(I,FF);:NEXTI
CF  9460 PRINTBL$;"[BLUE,SPC10,R
    VSON]C[RVSOFF]HANGE. [RVSON]
    D[RVSOFF]ELETE. [RVSON]N[RVS
    OFF]EXT.[SPC7,UP5]";
97  9470 GETA$:IFA$=""THEN9470
EF  9475 IFA$="N"THEN9410
D0  9480 IFA$="D"THEN14500
D1  9485 IFA$="C"THEN15500
BD  9490 RUN
0F  9500 PRINT"[CLR,DOWN3,GREEN,
    SPC12]INSERT BLANK TAPE."
D3  9510 PRINT"[DOWN2,SPC12]REWI
    ND THE TAPE TO"
B5  9520 PRINT"[DOWN2,SPC12]WHER
    E YOU WANT IT."
0B  9530 PRINT"[DOWN6,SPC9]PRESS
     ANY KEY WHEN READY.[DOWN4]"
DF  9535 GETA$:IFA$=""THEN9535
09  9536 IFA$=" "THENRUN
0D  9540 IFPEEK(52606)=32THEN957
    0
FC  9550 POKE45,134:POKE46,159:P
    OKE56,160:CLR
09  9560 POKE171,0:SYS40026
00  9570 SYS(680)"ADDRESS BOOK",
    2048,40960
1B  9580 RUN
4B  11000 PRINT"[CLR]";
AE  11005 PRINT"[PURPLE,CO10]"
71  11010 PRINT"[RVSON,PURPLE]SU
    RNAME[SPC3]"
B6  11020 PRINT"[PURPLE,CY10]"
CB  11030 PRINT"[PURPLE,CO10]"
D0  11040 PRINT"[RVSON,PURPLE]TI
    TLE[SPC5]"
E0  11050 PRINT"[PURPLE,CY10]"
B6  11060 PRINT"[PURPLE,CO10]"
19  11070 PRINT"[RVSON,PURPLE]FO
    RENAME/S"
12  11080 PRINT"[PURPLE,CY10]"
0C  11090 PRINT"[PURPLE,CO10]"
CE  11100 PRINT"[RVSON,PURPLE]AD
    DRESS[SPC3]"
1C  11110 PRINT"[PURPLE,CY10]"
EA  11120 PRINT"[PURPLE,CO10]"
40  11130 PRINT"[RVSON,PURPLE]AD
    DRESS[SPC3]"
05  11131 PRINT"[PURPLE,CY10]"
2E  11132 PRINT"[PURPLE,CO10]"
57  11133 PRINT"[RVSON,PURPLE]AD
    DRESS[SPC3]"
34  11134 PRINT"[PURPLE,CY10]"
E5  11135 PRINT"[PURPLE,CO10]"
52  11136 PRINT"[RVSON,PURPLE]AD
    DRESS[SPC3]"
37  11137 PRINT"[PURPLE,CY10]"
E6  11140 PRINT"[PURPLE,CO10]"
15  11141 PRINT"[RVSON,PURPLE]TE
    L. NO.  "
24  11150 PRINT"[PURPLE,CY10]"
4D  11160 RETURN
75  12000 FORI=1TONF:D$=D$(I):GO
    SUB13000:IP$(I)=B$:NEXTI
24  12080 PRINTBL$;"[BLUE,SPC10]
    IS THIS CORRECT ?  Y/N.";
C6  12090 GETA$:IFA$=""THEN12090
4A  12100 IFA$="N"THENRUN
72  12105 IFA$<>"Y"THEN12090
5A  12110 IFIP$(1)=""THENRUN
0F  12120 PP=PP+1:PRINT"[CLR,C5]
    10 PP=";PP
89  12130 FORI=1TO8:PRINT1000*I+
    PP-1"OP$("I","PP-1")="CHR$(3
    4);IP$(I);CHR$(34):NEXTI
A7  12140 PRINT"RUN"
52  12150 POKE631,19:FORI=1TO10:
    POKE631+I,13:NEXTI:POKE198,1
    1:END
16  13000 N=KW:B$=""
8E  13020 PRINTD$;X$;B$;CU$
36  13030 GETA$:IFA$=""THEN13030
9A  13040 IFASC(A$)=13THENPRINTD
    $;X$;B$;" ":RETURN
FE  13050 IFASC(A$)=20THENGOSUB1
    5000:GOTO13020
46  13060 IFASC(A$)=34ORASC(A$)=
    44ORASC(A$)=59THEN13020
F7  13061 IFASC(A$)=133THENB$="M
    R & MRS ":GOTO13095
B2  13070 IFASC(A$)<32ORASC(A$)>
    127THEN13020
AF  13072 IFA$="_"THENRUN
25  13075 IFN=21THEN13020
75  13080 N=N+1:B$=B$+A$
85  13095 GOTO13020
8A  14000 D$=D$(1):GOSUB13000:SK
    $=B$:FF=1:IFSK$=""THENRUN
96  14010 FR=0:FORI=FFTOPP
53  14020 IFSK$=OP$(1,I)THENFF=I
    :I=PP:FR=1
C6  14025 NEXTI
C9  14030 IFFR=0THENPRINT"[CLR,D
    OWN,CYAN,SPC13]NAME NOT FOUN
    D.":FORDL=1TO3000:NEXT:RUN
8D  14040 FORI=1TO8:D$=D$(I):PRI
    NTD$(I);X$;OP$(I,FF);:NEXTI
4B  14045 PRINTBL$;"[BLUE,SPC10]
    IS THIS THE RIGHT ONE ?  Y/N
    .";
30  14046 GETA$:IFA$=""THEN14046
A5  14047 IFA$="N"THENGOSUB17000
    :FR=0:FF=FF+1:GOTO14010
E5  14048 IFA$<>"Y"THEN14046
80  14050 PRINTBL$;"[SPC10,RVSON
    ]C[RVSOFF]HANGE. [RVSON]D[RV
    SOFF]ELETE. [RVSON]E[RVSOFF]
    XIT. [RVSON]P[RVSOFF]RINT. [
    UP5]";
3E  14060 GETA$:IFA$=""THEN14060
DF  14070 IFA$="C"THEN15500
7B  14075 IFA$="P"THEN14600
4E  14080 IFA$="D"THEN14500
CB  14090 RUN
E7  14440 FORI=1TO8:D$=D$(I):PRI
    NTD$(I);X$;OP$(I,FF);:NEXTI
40  14500 PRINT"[CLR]";:FORI=1TO
    NF:PRINT1000*I+FF:NEXTI
28  14510 PRINT"RUN"
33  14520 POKE631,19:FORI=1TO10:
    POKE631+I,13:NEXTI:POKE198,1
    1:END
4F  14600 IFQW=1THEN14620
31  14605 QW=1:PRINTBL$;X$;SP$:G
    OSUB9200:OPEN1,4
F6  14610 GOSUB11000
B7  14615 FORI=1TO8:D$=D$(I):PRI
    NTD$(I);X$;OP$(I,FF);:NEXTI
74  14620 GOSUB49600:GOTO14050
15  15000 IFN=1THENB$="":RETURN
F5  15020 N=N-1:B$=LEFT$(B$,N-1)
    :PRINTD$;X$;B$;"[CD] ":RETUR
    N
69  15500 NA=0:CE=0
7A  15510 NA=NA+1:IFNA=NF+1THEN1
    5700
38  15515 PRINTBL$;"[BLUE,SPC10]
    CHANGE THIS LINE ? Y/N. ";
C2  15520 PRINTD$(NA);X$;">";SP$
    ;
D7  15530 FORJ=1TO100:NEXTJ
14  15540 PRINTD$(NA);X$;OP$(NA,
    FF)
D8  15545 FORJ=1TO100:NEXTJ
E4  15550 GETA$:IFA$=""THEN15520
2F  15560 IFA$="N"THEN15510
49  15565 IFA$<>"Y"THEN15550
28  15566 PRINTBL$;"[BLUE,SPC10]
    TYPE IN THE DETAILS.[SPC9]";
8A  15570 D$=D$(NA)
79  15572 PRINTD$(NA);X$;SP$;
45  15580 GOSUB13000
C7  15590 PRINTBL$;"[BLUE,SPC10]
    IS THIS CORRECT ?  Y/N. ";
1D  15600 GETA$:IFA$=""THEN15600
A5  15602 IFA$="Y"ANDKW>1THENRET
    URN
4A  15603 IFA$="N"ANDKW>1THEN155
    66
29  15610 IFA$="Y"THENCE=1:OP$(N
    A,FF)=B$:GOTO15510
4A  15615 IFA$<>"N"THEN15600
D1  15620 PRINTD$(NA);X$;"[SPC29
    ]";
49  15630 PRINTD$(NA);X$;OP$(NA,
    FF);:NA=NA-1:GOTO15510
```

```
2B   15700 IFCE=0THENRUN
24   15702 IFOP$(1,FF)=""THENRUN
C8   15705 PRINT"[CLR]";:FORI=1TO
     NF
93   15710 PRINT1000*I+FF;"OP$("I
     ","FF")=";CHR$(34);OP$(I,FF)
     ;CHR$(34)
70   15720 NEXTI
33   15730 PRINT"RUN"
3E   15740 POKE631,19:FORI=1TO10:
     POKE631+I,13:NEXTI:POKE198,1
     1:END
3D   17000 FORS=1TONF:PRINTD$(S);
     X$;SP$;:NEXTS:RETURN
A7   18000 FORI=1TOPP
72   18010 IFOP$(1,I)=""THENOS=I:
     I=PP
93   18050 NEXTI
66   18060 FO=0:FORI=OS+1TOPP
60   18065 IFOP$(1,I)<>""THENNS=I
     :I=PP:FO=1
A7   18070 NEXTI
08   18075 IFFOTHEN18090
72   18076 PRINT"[CLR]10 PP=";OS+
     1:PRINT"21000 REM"
23   18080 PRINT"38000 KB=0":PRIN
     T"RUN":POKE631,13:POKE632,13
     :POKE633,13:POKE634,13:POKE6
     35,13
AF   18085 POKE198,5:END
4D   18090 PRINT"[CLR]21000 OS="O
     S":NS="NS
BD   20010 FORI=1TO8
E6   20020 PRINT1000*I+OS;"OP$("I
     ","OS")="CHR$(34);OP$(I,NS);
     CHR$(34)
4E   20025 NEXTI
C9   20030 PRINT"RUN 21000"
F9   20040 POKE631,19:FORI=1TO10:
     POKE631+I,13:NEXTI:POKE198,1
     1:END
D5   21000 REM
25   21005 PRINT"[CLR]";:FORI=1TO
     8
96   21010 PRINT1000*I+NS
35   21020 NEXTI
0A   21030 PRINT"RUN"
6D   21040 POKE631,19:FORI=1TO10:
     POKE631+I,13:NEXTI:POKE198,1
     1:END
53   38000 KB=0
D0   38010 KW=1:IFKB=1THEN18000
C4   38020 D$(1)="[HOME,DOWN]":D$
     (2)=D$(1)+"[DOWN3]":D$(3)=D$
     (2)+"[DOWN3]":D$(4)=D$(3)+"[
     DOWN3]"
3F   38022 D5$=D$(4)+"[DOWN3]"
5A   38030 D$(5)=D$(4)+"[DOWN3]":
     D$(6)=D$(5)+"[DOWN3]":D$(7)=
     D$(6)+"[DOWN3]":CU$="[CD]"
99   38035 D$(8)=D$(7)+"[DOWN3]"
6E   38040 BL$="[HOME,DOWN24]":X$
     ="[RIGHT11,CS]"
44   38050 NF=8
FD   38060 SP$="[SPC28]"
C4   38070 RETURN
C9   49400 OPEN1,4:FF=0:GOSUB1100
     0
92   49410 FF=FF+1
8E   49420 IFFFF=PPTHENPRINT"[CLR
     ,DOWN,RED,SPC14]END OF BOOK"
     :FORDL=1TO3000:NEXT:RUN
CC   49430 IFOP$(1,FF)=""THEN4941
     0
1A   49440 GOSUB17000
C0   49450 FORI=1TONF:PRINTD$(I);
     X$;OP$(I,FF);:NEXTI
4C   49460 PRINTBL$;"[C5,SPC10,RV
     SON]N[RVSOFF]EXT. [RVSON]P[R
     VSOFF]RINT.";
E0   49470 GETA$:IFA$=""THEN49470
29   49475 IFA$="N"THEN49410
69   49476 IFA$=" "THENRUN
4E   49480 IFA$<>"P"THEN49470
FF   49485 IFLEN(OP$(1,FF))+LEN(O
     P$(2,FF))+1<21THEN49600
0E   49490 PRINTBL$;X$;"[RED,LEFT
     ]TITLE TOO LONG FOR LABEL.";
     :FORDD=1TO3000:NEXTDD
DD   49494 X=20-LEN(OP$(1,FF))-1
E8   49495 PRINTBL$;"RE-DO TITLE.
      MAX -";X;"[LEFT]  CHARACTER
     S. ";
1F   49500 PRINTD$(2);X$;SP$;
A4   49510 KW=LEN(OP$(1,FF))+2:B$
     ="":NA=2:GOSUB15570
ED   49600 FORS=1TO7
FE   49610 IFS<>1THEN49640
DF   49615 IFKW=1THENP$=OP$(2,FF)
     +"[SSPC]"+OP$(1,FF):GOTO4963
     0
67   49620 KW=1:P$=B$+" "+OP$(1,F
     F)
CF   49630 GOTO49660
37   49640 IFS=2THENS=4
BC   49650 P$=OP$(S,FF)
E2   49660 PRINT#1,CHR$(1);P$
B8   49670 NEXTS:PRINT#1:PRINT#1:
     IFQW=1THENRETURN
DC   49675 POKE198,0:GETA$:IFA$="
     "THENRUN
0B   49680 GOTO49410
DA   55000 POKE198,0:WAIT198,1
79   60000 GETA$:IFA$=""THEN60000
72   60010 PRINTASC(A$):GOTO60000
```

American Club Sports



Do you remember a game called *Indoor Sports*, that received rave reviews when it was imported into the UK by the now defunct Advance Software? Well, now the game's US publishers Mindscape has set up shop on this side of the pond, and has released its sequel *American Club Sports*.

The game consists of six sports that can be played individually against the computer, or with up to three friends. A title program is loaded in first, and from there the other games are selected. On the tape version – the review copy – this means remembering the counter position for each sport or a long wait. First up is Foosball, which is table football to you and me. It's the bar type that's played in arcades, where you have to spin the players to kick the ball into your opponent's goal.

Perhaps more unusual is the inclusion of two types of billiards – the English and Carom variants. Most people will know the English game, so I'll concentrate on the variant, in which there are no pockets and you have to declare whether your effort will be a shot or a safety shot. If you're wrong, your opponent gets a point. This means that points are scored via cannons, and the game continues until a set score is reached.

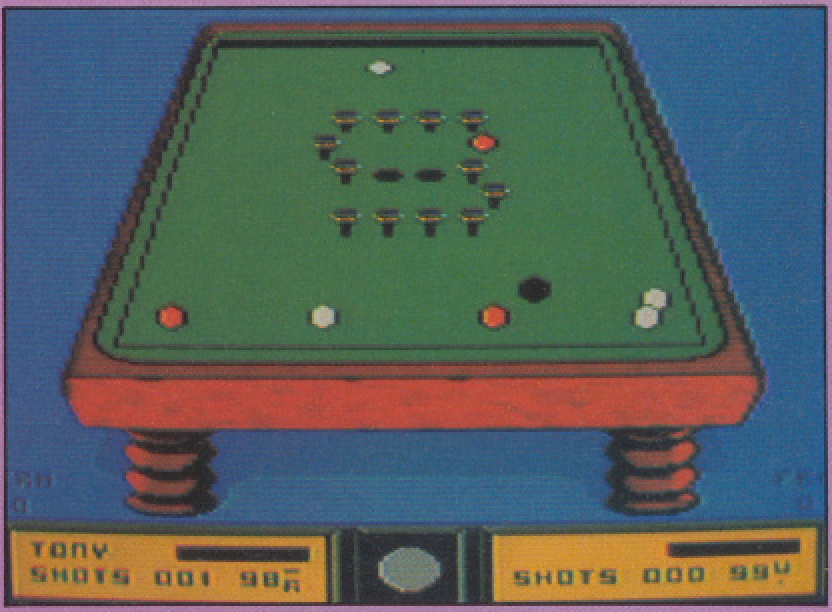
Pinball is a welcome addition, as I don't think there are enough pinball games around. This one features a 3D table, and you can even alter factors such as the angle of the table to speed up the game, the tilt sensor and point thresholds at which you'll gain extra balls to make the game as easy or as difficult as you want it to be.

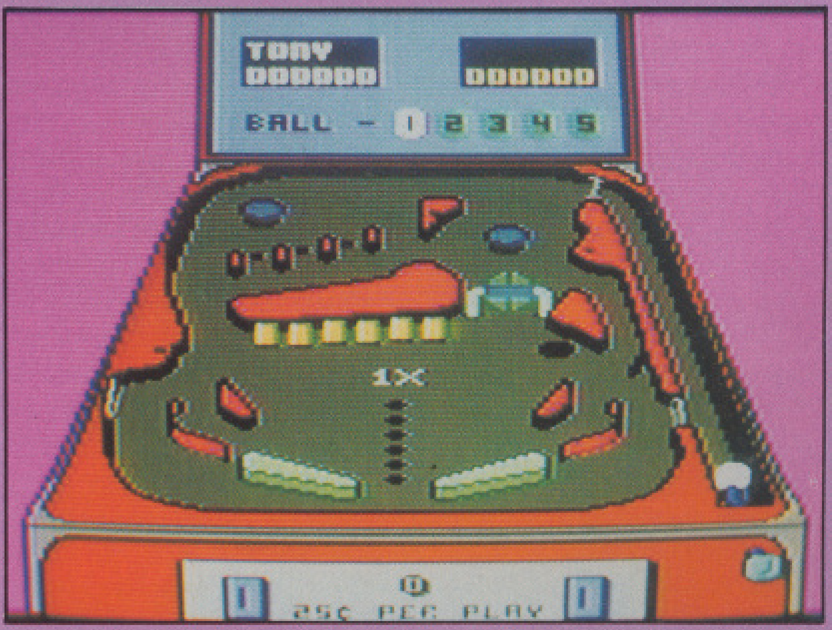
Baseball is a strange game in which you must roll balls up a ramp and into one of the scoring rings at the top of the screen. The big points are scored by landing in the centre rings, but the pressure is piled on if you play either against the clock or with a set number of balls.

Crazy Pool is weird – it's played on a table that's a cross between a pool table and a pinball machine. The normal six pockets have vanished and are replaced by two in the centre of the table. Unfortunately, these are surrounded by pinball style bumpers arranged in an S shape, making potting difficult and totally altering the tactics of the game.

Finally, there's a change from aiming balls into holes, rings, goals and pockets, as the sixth event is a shooting gallery in which you blow away ducks, rabbits, birds, foxes, frogs and fish to rack up the points. Once again, the choice is either to fight against the clock or to shoot with a limited number of bullets. Either way, this completes an interesting compilation of club sports. *Indoor Sports* was a big hit, and this is sure to have some success. My favourite events were pinball and billiards.

Touchline:
Title: *American Club Sports.* ***Supplier:*** *Mindscape, PO Box 1019, Lewes, East Sussex, BN8 4DW.* ***Price:*** *£9.99 cass, £14.99 disk.*





Speedball

Duncan Evans reviews the game that makes Rollerball look like Tiddly-winks



The Bitmap Brothers originally shot to fame with *Xenon*, which is just now becoming available on the 8-bit formats, then turned their hands to a futuristic sports game on the ST and Amiga. *Speedball* was greeted with great critical acclaim on its release, and has now found its way onto the C64.

The question is, has it survived the transition well? I'm glad to be able to report that it has indeed, and retains virtually all of the flavour of the 16-bit versions.

Speedball is a game of organised violence. Two teams of five players (including a goalkeeper), race up and down a vertically scrolling pitch, trying to ram a steel ball into a thin goal slot while fending off the opposition.

The view is from above, and if the ball gets thrown above head height (when it becomes impossible to score), then it grows in size as it gets higher, before decreasing as it drops earthwards.

Basically, in this game anything goes, so if someone from the other team whacks you in the kidneys and takes the ball, just dust yourself off, go after him, push his nose through the back of his head, and then retrieve the ball. Fighting the opposition is an integral part of the game as it reduces their stamina, thus making them slower and more susceptible to foul play.

Adorning the pitch are bumper-like obstructions, off which the ball ricochets. There are a number of different pitch combinations, some of them making scoring quite difficult. From the main menu you can either play a one-off game against a friend, or compete against the computer in a league or knockout competition.

When playing in a competition (the league can be set from 10-100 weeks) there is another important factor to consider – Icons. When playing a game, some of these icons have an immediate effect, like freezing the opponent, or slowing them down, or giving you an unstoppable ball, but there are others which are collectable.

These are very important, in that you can buy certain services or improvements at the end of the game, even bribe the timing official into letting you have another minute of injury time. Most of them improve the statistics of your team's stamina, skill or strength, but once these have reached maximum then you'll be buying goals, bribing their coach, making the icons last longer, and all sorts of other dirty tricks.

If you wanted to compare *Speedball* to anything it would be *Rollerball*, but what I reckon proved the inspiration for the game was a cartoon strip from many years ago in the comic *Battle-Action*, called *Spinball*. Except for the goal size, the similarity is quite striking.

COMMENT

Given that the C64 has excellent sound and scrolling facilities, it was only a matter of whether the detail of the original game could be implemented. Obviously there is some small loss, but not much, and more importantly the gameplay is still there. If anything this version is more playable, because when playing the worst computer opponents, the goalkeeper's like a lobotomy victim, which is good news for novice players though it does get very tough against the highly rated teams.

The graphics are on the whole pretty good, and the music, while nothing special, is certainly better than the ST version's was. The good thing about the length of each individual game is that it is long enough not to guarantee the first scorer victory, and yet short enough to let you play a few on the trot in a league season. The league is a great idea, though you're very unlikely to win it if you play over 10 weeks. A minimum of 30 is required to collect enough tokens to bring your team up to full strength and still have enough games left to mount a challenge.

Speedball is a fast and furious action game, and with the league option, has enough playability to keep you going long after the initial glamour has worn off.

Touchline:
Title: *Speedball.* ***Supplier:*** *Mirrorsoft, Irwin House, 118 Southwark Street, London SE1 0SW.* ***Tel:*** *01-928 1454.* ***Price:*** *£9.95.*





Circus Attractions

Always the clown, Duncan Evans fools around with Rainbow Arts in the big top

Circus Attractions is the latest attempt by Rainbow Arts to get you to buy German, and after the rather poor *Grand Monster Slam*, shows some semblance of a return to form.

Well, I didn't think I'd be able to stomach another games compendium, but *Circus Attractions* shows that if you flog a dead horse hard enough, it might not get up and run around, but will at least twitch a lot.

Yep, a collection of games this most certainly is, set in the big top, and featuring all the fun of the fair. I never did like the circus, but this game contains a number of passable attempts to inject humour into the procedings which enliven the otherwise tedious quest for points and a high score.

Among the death-defying feats you'll be attempting are trampolining, tightrope walking (gulp!), juggling, knife throwing, and last but not least, being a jumping clown. Five events don't make much of a circus show in my book, but at least you have the choice of practicing them before inflicting your ineptitude on the public.

Trampolining then, has you bouncing up and down, monitored by a "visitor interest" column. When this sinks to zero, the audience starts throwing fruit, and who can blame them, because bouncing up and down on a trampoline isn't that exciting. You're expected to perform forward and backward somersaults, but I didn't bother perfecting my technique on this event. Dusting off the dirt accumulated by repeatedly hitting the deck, I began to climb bravely up to the tightrope.

In this a pretty young lady precariously balances on the rope high above the ground, and makes her way along automatically. Rather her than me really, especially since you're supposed to perform a scissor kick, a backward somersault and a hand stand. As soon as I started, the poor girl wobbled dramatically and disappeared earthwards.

For complexity, the juggling event takes the biscuit. Not only are you required to juggle up to six balls in order, but you're supposed to juggle a club as well, volley a balance ball and leap up in the air when a clown comes racing towards you on a motorbike. It ain't easy, believe me.

I liked the next event – knife-throwing – the best, probably because as my doctor once told me, I'm a homicidal maniac. Strapped to a rotating disc is the latest lovely young assistant. Simply aim a crosshair and let fly. What fun.

There are other targets beside the girl, but she's the main one. You must also watch out for sticks of dynamite being handed to you by your other assistant. Watching people curse as you stick them is great fun, and this is the one section I practiced religiously until I could hit someone with every throw.

The final event of the night is the jumping clowns, which features three clowns leaping from seesaw to seesaw collecting bonuses in mid-air and avoiding the ghosts that lurk up there.

COMMENT

I'm surprised that anyone bothers to try milking a very tired formula, but to give Rainbow Arts its due, a couple of these events are fairly novel, and quite entertaining. The graphics are very good all round – only the tightrope walking is disappointing.

In this, the lady crossing the big top is drawn very small, but worst of all, she merges with the background, making it very difficult to see whether she's about to fall to a sticky end or is making short work of it all.

Of all the events, there was only one that I actually enjoyed (knife-throwing). The others were ones that had to be played simply because I was reviewing the game. If you are a fan of event-style games, then you'll probably get more out of it than I did.

It's interesting that in the two player mode you have to cooperate to score, rather than directly compete, which may or not may be a good thing depending on your viewpoint.

Circus Attraction is high on gloss (with the sound effects and music done in a zany circus vein), but a little short on the gameplay front. With only five events, it doesn't take too many dull ones to make the whole thing fail. Trying before you buy is essential with this one.

Touchline

Title:*Circus Attractions.* ***Supplier:****Rainbow Arts, Hansaalle 201, 4000 Dusseldorf 11, West Germany.* ***Tel:****0211-596764.* ***Price:****£9.95*

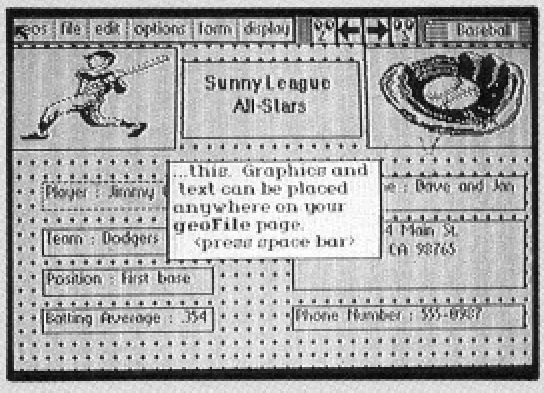
Through The R

Paul Eves puts the latest update of the excellent Geos package through its paces

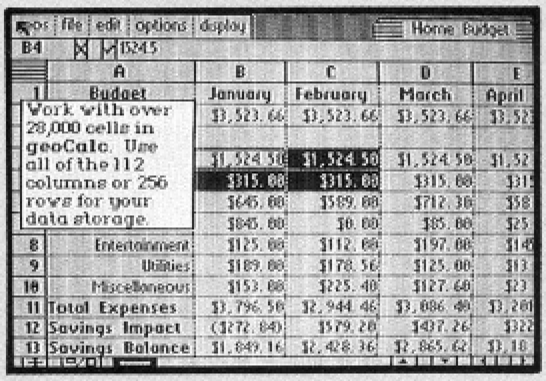
Being a neat and tidy sort of person, I welcomed the opportunity of trying out this latest Geos application. However, before I go any further, I must be perfectly honest and say that I don't normally use these kind of programmes. Yes, they do look very nice, and they also do a very competent job. It's just that somehow, a little of the magic of using a computer seems to vanish when you use this sort of package – at least for me it does.

For those that do not yet know, GEOS stands for 'Graphic Environment Operating System', in other words, a system that offers windows, icons and pull-down menus. Anyway, my first thought on taking out the manuals (yes, manuals, as in more than one) was that if I'd wanted to read *War and Peace* I would have bought it. I needn't have worried though – a quick flip through the pages put my mind at ease. The books are well thought out, and explain everything in great detail, so that even a complete novice to the world of Geos will soon feel at home.

The main problem with a package like Geos is where to begin – there's so much on offer. You don't want to miss anything out, but at the same time you don't want to go into too much detail and rewrite the programme. So I've decided to tackle this particular review in a more systematic way. First, I'll list all the applications on the disks, then briefly go over some of the finer aspects of some of the applications. Obviously I can't cover everything in the limited space of this review, but by the end you should have a fairly good idea of the package's potential.



There are three disks in the package, each being double sided. The first is the main systems disk, with the reverse side given up for demos. Disk number 2 is the back-up systems disk, with a few applications on the reverse. The final disk contains the 'Write Utilities' and a spell checker. By the way, there is in fact a fourth disk, a demonstration of the Quantumlink.



As you may or may not know, Quantumlink is the Stateside equivalent to our Compunet).

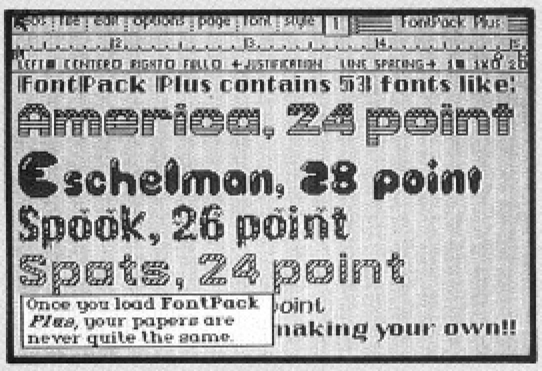
The programmes that make up this enhanced system are as follows:-

1) The main GEOS desktop
2) geoPaint
3) geoWrite 2.1
4) geoSpell
5) geoMerge
6) geoLaser
7) Text Grabber
8) Paint Drivers
9) Desk Accessories

The desk accessories are as follows:-

1) The Calculator
2) The Preference Manager
3) The Alarm Clock
4) The Note Pad
5) The Pad Colour Manager
6) The Photo manager
7) The Text Manager

As I think you'll agree, that's a pretty impressive list. So what exactly does each one do? Read on...



Desktop

This is the main driving force of the system. From the desktop you can perform all your file handling procedures. An extra bonus with the Version 2.0 is the use of cursor keys for pulling down menus and making selections. Indeed, you have a whole range of keyboard shortcuts. Whenever you make up a working disk, it is advisable to include the Desktop on each one, along with whatever else you may require.

Geopaint

I have never been one for paint packages of any description, I suppose partly because I am not artistic by nature. Secondly, I have always found them to be rather long-winded and awkward to use. I must confess, however, that although Geopaint is fairly complex in its functions, I found it was actually fun to use.

The options in this section of the package seem endless. You can create images using special measurement and constraining tools. You have access to 32 patterns and brush styles, overlay can be achieved and text may be mixed with images drawn, you can stretch images or zoom in on them, and print your creations on a numerous list of different printers. For those very small increments needed, you can use the

Round Window

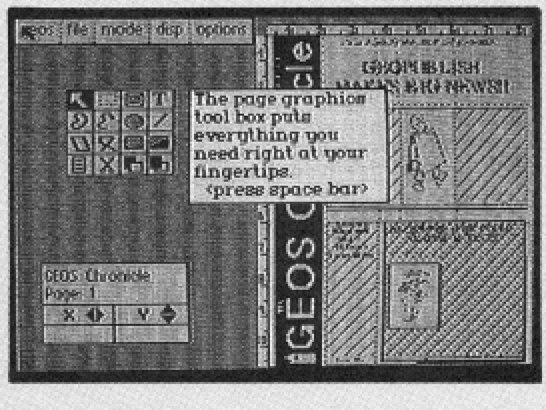
cursor keys instead of the joystick or mouse, for more accurate placement.

Geopaint works in either 40 or 80 column modes, and is interchangeable while working within it. However, you can only work with colour in the 40 column mode. The advantage of working in 80 column mode though is obvious – you can see the whole of your work area and plan accordingly. Once you have the main parts drawn, switch to 40 column mode and you can then work in more detail.

Geowrite 2.1

Most of us who use wordprocessors tend to stick with the one we know best. For example, for some time I only ever used Easyscript. I know it's pretty primitive compared to most, but I knew it inside out and back to front. However, after many months of badgering from the Editor, I relented and now use either Superscript or Paperclip II. (Hip-Hip Hooray...Ed!).

I never really used Geowrite on earlier Geos packages, but having used the Geowrite 2.1, I now think that maybe I was missing out. This wordprocessor, like everything else in the system, is pretty comprehensive. You have options to alter your document's dimensions, change the writing window, even the ability to have different fonts and styles. The fact that you can mix your creations from Geopaint with your text is most useful. You can even add the date and time to your page headers and footers. The usual Copy, Cut, Paste, Move Text and Set



Tabs options are all available. You can search for and search and replace text, not only single words but whole phrases.

In conjunction with Geowrite 2.1, there are other related applications – Geospell, Geomerge, Text Manager, and Text Grabber. I don't think I need explain Geospell and Geomerge in any great detail. They are essentially like most other spell checkers and document mergers. The text manager is like a temporary storage area – you can copy text into what is known as albums for future recall. The text manager works in the same way as Glossaries from other wordprocessing packages.

The one really nice feature is the Text Grabber. This application allows you to get a document that was created on some other Commodore supportive W/P, then convert it to Geowrite format. The original document will remain unchanged.

One feature I nearly forgot is the Paint Drivers. These drivers allow you to create special effects within your Geowrite document. You can have things like headlines, newspaper-type column formats including graphics, special border designs, etc.

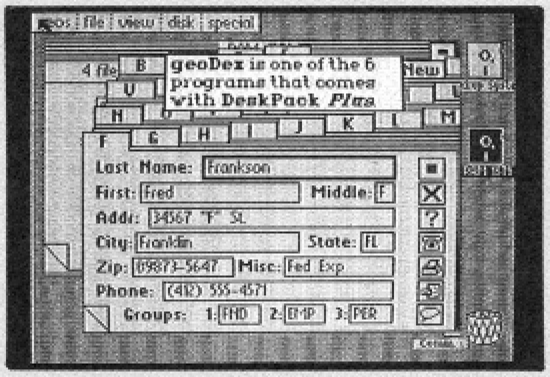
Desktop Accessories

The calculator allows you to do your calculations while running any of the Geos applications. The results you get may then be placed into the text scrap so that you can recall them later, an example would be if you wanted to include them in an invoice you were preparing.

The Preference Manager enables you to set up your own working environment. That is to say the colours you want to work with, the colour of the pointer and the shape of the pointer. You can change the speed of the pointer and set up the date and time.

The Alarm Clock can be used to call up the current time, providing it has been set. You can set the correct time and also set the alarm. Once the clock and/or the alarm are set, it doesn't matter which application you are in, the clock stays active.

As its name suggests, the note pad is used for keeping track of bits of information you may wish to refer back to. You have access to the note pad no matter which application you are presently in. The note pad can store up to 127 pages, and each page is capable of roughly 250 characters.



Like the note pad, Photo Manager is used for storing graphic images. The files stored are called albums. An album may contain up to 60 pages of images. Therefore, you could have albums set aside for specific images, for example, you might have a graph album, a pie chart album, etc.

That's just about it. I've really only skimmed the surface of this package's capabilities. There's so much you could say about each application that you'd need a whole magazine to do it real justice. I have tried, in my own small way, to point out some of the finer qualities. The only real way for you to appreciate what Geos has to offer is to go and buy it. I've included a couple of examples of the sort of things you can produce. I hope you like them.

Disk Scrambler

Protect your disks from prying eyes with this Disk Scrambler

By S. T. Burke



Disk Scrambler enables you to encode or decode the contents of any block or blocks on your disk. There are just two basic options open to you:

SCRAMBLE – this encodes the specified area of the disk. You may do individual blocks, a series of blocks or the entire disk. The scrambler alters the code on the disk using a different technique for each two-character code entered by the user.

DESCRAMBLE – this is opposite to SCRAMBLE. The specified area of the disk is decoded using the *same* two-character code entered previously for that area of the disk. Please note that if a different two character code is used, the descrambling will not be successful.

If you happen to enter a wrong code on the descramble option, refer to the troubleshooting guide.

The program is straightforward and self explanatory. Type in the listing and save it before running. Once again, just follow the on-screen options.

Troubleshooting

If the program fails to work as you think it should, first reset the computer, then reload the program, and try a few times on a blank disk.

If it still fails, check your listing thoroughly, (you may have made a mistake). If you find no errors, check out your hardware.

If you scramble a disk a number of times, or attempt to descramble it a number of times, and you cannot restore your disk, then you must descramble it with *all* the codes ever used on that particular disk since it worked (they don't have to be in any set order).

Please note that the author cannot be held responsible for any disks that may become corrupted beyond repair by the use of this program.

Finally, an interesting aspect of the program is this – if you want to play a trick on someone, use the two character code NS. The program will appear to work correctly, but in fact does nothing!

LISTINGS

DISK SCRAMBLER

```
59   10 POKE53280,0:POKE53281,11:
     POKE53265,0
BA   20 PRINT"[CLR,REV H,REV SN,R
     VSON,C5,SM,SPC31,C*]":CLR:DI
     MT(35):POKE649,1
E0   30 PRINT"[RVSON] [C8,SPC32]"
     :FORT=1TO35:READT(T):NEXT
67   40 PRINT"[RVSON,C5] [C8,SPC9
     ]DISC SCRAMBLER[SPC9]":FORA=
     679TO709:READP:POKEA,P:NEXT
E6   50 PRINT"[RVSON,C5] [C8,SPC9
     ,CT14,SPC9]":OPEN15,8,15,"I"
     :CLOSE15
5B   60 PRINT"[RVSON,C5] [C8] [WH
     ITE]0.[C8] SCRAMBLE WHOLE DI
     SC[SPC9]"
04   70 PRINT"[RVSON,C5] [C8,SPC3
     0]"
B8   80 PRINT"[RVSON,C5] [C8] [WH
     ITE]1.[C8] SCRAMBLE INDIVIDU
     AL TRACK[SPC3]"
68   90 PRINT"[RVSON,C5] [C8,SPC3
     0]"
AB   100 PRINT"[RVSON,C5] [C8] [W
     HITE]2.[C8] SCRAMBLE RANGE O
     F TRACKS[SPC4]"
3C   110 PRINT"[RVSON,C5] [C8,SPC
     32]"
14   120 PRINT"[RVSON,C5] [C8] [W
     HITE]3.[C8] SCRAMBLE INDIVID
     UAL BLOCK[SPC3]"
A0   130 PRINT"[RVSON,C5] [C8,SPC
     32]"
A3   140 PRINT"[RVSON,C5] [C8] [W
     HITE]4.[C8] SCRAMBLE RANGE O
     F BLOCKS[SPC4]"
14   150 PRINT"[RVSON,C5] [C8,SPC
     32]"
65   160 PRINT"[RVSON,C5] [C8] [W
     HITE]5.[C8] DESCRAMBLE WHOLE
      DISC[SPC7]"
B8   170 PRINT"[RVSON,C5] [C8,SPC
     32]"
1A   180 PRINT"[RVSON,C5] [C8] [W
     HITE]6.[C8] DESCRAMBLE INDIV
     IDUAL TRACK "
4C   190 PRINT"[RVSON,C5] [C8,SPC
     32]"
```

```
15  200 PRINT"[RVSON,C5] [C8] [W
    HITE]7.[C8] DESCRAMBLE RANGE
     OF TRACKS   "
EA  210 PRINT"[RVSON,C5] [C8,SPC
    32,C5]m[SPC4"
12  220 PRINT"[RVSON] [C8] [WHIT
    E]8.[C8] DESCRAMBLE INDIVIDU
    AL BLOCK [RVSOFF,C5,C*,RVSON
    ,C8](C)YC"
A3  230 PRINT"[RVSON,C5] [C8,SPC
    32,RIGHT,C5]m[SPC3"
59  240 PRINT"[RVSON] [C8] [WHIT
    E]9.[C8] DESCRAMBLE RANGE OF
     BLOCKS  [RIGHT,RVSOFF,C5,C*
    ,RVSON,C8]1989"
88  250 PRINT"[RVSON,C5] [C8,SPC
    32]"
C2  260 PRINT"[C5,C*,RVSON,C8,SP
    C7]PLEASE MAKE CHOICE[SPC7,H
    OME]":POKE53265,27:POKE198,0
BF  270 GETA$:IFVAL(A$)=0ANDA$<>
    "0"THEN270
66  280 A=VAL(A$):A$=""
0F  290 IFA>4THENA$="DE":A=A-5
4F  300 POKE53265,0:POKE53281,11
    :PRINT"[CLR,C5]CODE:*[LEFT]"
    ;:POKE53265,27
9C  310 GETC1$:IFC1$=""THEN310
7B  320 PRINT"[RVSON] [RVSOFF]*[
    LEFT]"
58  330 GETC2$:IFC2$=""THEN330
F7  340 NS=-1:CD$=C1$+C2$:PRINT"
    [CLR]";
DC  350 IFCD$="NS"THENNS=0
29  360 POKE251,ASC(C2$):POKE252
    ,ASC(C1$)
8E  370 ONA+1GOTO660,710,740,800
    ,840
BC  380 POKE53280,0:POKE53281,0:
    POKE53265,0
CB  390 PRINT"[CLR,REV H,REV SN,
    RIGHT,RVSON,CYAN,S\,SNS]  (C
    )YOUR COMMODORE 1989  [SMS,C
    *]"
1F  400 PRINT"[BLUE]cccccccccccc
    cccccccccccccccccccccccccccc
    cc[WHITE]cccc[BLUE]ccc[WHITE
    ]cccccc[BLUE]ccc[WHITE]cccc[B
    LUE]ccc[WHITE]";
A9  410 PRINT"cccc[BLUE]cccccccc
    cccccc[WHITE]c[BLUE]ccc[WHIT
    E]c[BLUE]cccc[WHITE]c[BLUE]c
    ccc[WHITE]c[BLUE]cccccc[WHIT
    E]c[BLUE]cccccccccccccccccc[
    WHITE]c[BLUE]";
C7  420 PRINT"ccc[WHITE]c[BLUE]c
    ccc[WHITE]c[BLUE]cccc[WHITE]
    c[BLUE]cccccc[WHITE]c[BLUE]c
    cccccccccccccccc[WHITE]c[BL
    UE]ccc[WHITE]c[BLUE]cccc[WHI
    TE]c[BLUE]ccccc[WHITE]ccc";
A9  430 PRINT"[BLUE]ccc[WHITE]c[
    BLUE]cccccccccccccccccc[WHIT
    E]c[BLUE]ccc[WHITE]c[BLUE]cc
    cc[WHITE]c[BLUE]cccccccc[WHI
    TE]c[BLUE]cc[WHITE]c[BLUE]cc
    cccccccccccc";
CF  440 PRINT"ccc[WHITE]c[BLUE]c
    cc[WHITE]c[BLUE]cccc[WHITE]c
    [BLUE]cccccccc[WHITE]c[BLUE]
    cc[WHITE]c[BLUE]cccccccccccc
    cccccc[WHITE]cccc[BLUE]ccc[W
    HITE]ccccc[BLUE]cc[WHITE]";
F1  450 PRINT"cccc[BLUE]ccccc[WHI
    TE]cccc[BLUE]cccccccccccccccc
    cccccccccccccccccccccccccccc
    cccccccccccc[RVSON,C7]c";
15  460 PRINT"c[RVSOFF,BLUE]cc[R
    VSON,C7]cc[RVSOFF,BLUE]c[RVS
    ON,C7]cc[RVSOFF,BLUE]ccc[RVS
    ON,C7]c[RVSOFF,BLUE]cc[RVSON
    ,C7]cc[RVSOFF,BLUE]c[RVSON,C
    7]cc[RVSOFF,BLUE]c[RVSON,C7]
    cc[RVSOFF,BLUE]cc[RVSON,C7]c
    [RVSOFF,BLUE]ccc[RVSON,C7]cc
    c[RVSOFF,BLUE]c[RVSON,C7]cc[
    RVSOFF,BLUE]";
9E  470 PRINT"cccc[RVSON,C7]c[RV
    SOFF,BLUE]ccc[RVSON,C7]c[RVS
    OFF,BLUE]ccc[RVSON,C7]c[RVSO
    FF,BLUE]c[RVSON,C7]c[RVSOFF,
    BLUE]c[RVSON,C7]c[RVSOFF,BLU
    E]c[RVSON,C7]c[RVSOFF,BLUE]c
    [RVSON,C7]c[RVSOFF,BLUE]c[RV
    SON,C7]c[RVSOFF,BLUE]c[RVSON
    ,C7]c[RVSOFF,BLUE]c[RVSON,C7
    ]c[RVSOFF,BLUE]c[RVSON,C7]c[
    RVSOFF]";
F0  480 PRINT"[BLUE]c[RVSON,C7]c
    [RVSOFF,BLUE]ccc[RVSON,C7]c[
    RVSOFF,BLUE]ccc[RVSON,C7]c[R
    VSOFF,BLUE]c[RVSON,C7]c[RVSO
    FF,BLUE]cccc[RVSON,C7]c[RVSO
    FF,BLUE]cc[RVSON,C7]c[RVSOFF
    ,BLUE]ccc[RVSON,C7]cc[RVSOFF
    ,BLUE]cc[RVSON,C7]ccc[RVSOFF
    ,BLUE]c[RVSON,C7]c[RVSOFF,BL
    UE]c[RVSON,C7]";
25  490 PRINT"c[RVSOFF,BLUE]c[RV
    SON,C7]c[RVSOFF,BLUE]c[RVSON
    ,C7]cc[RVSOFF,BLUE]cc[RVSON,
    C7]c[RVSOFF,BLUE]ccc[RVSON,C
    7]cc[RVSOFF,BLUE]cc[RVSON,C7
    ]c[RVSOFF,BLUE]cccccc[RVSON
    ,C7]c[RVSOFF,BLUE]c[RVSON,C7
    ]c[RVSOFF,BLUE]ccc[RVSON,C7]
    c[RVSOFF,BLUE]c[RVSON,C7]c[R
    VSOFF,BLUE]c";
AC  500 PRINT"[RVSON,C7]c[RVSOFF
    ,BLUE]c[RVSON,C7]c[RVSOFF,BL
    UE]c[RVSON,C7]c[RVSOFF,BLUE]
    c[RVSON,C7]c[RVSOFF,BLUE]c[R
    VSON,C7]c[RVSOFF,BLUE]c[RVSO
    N,C7]c[RVSOFF,BLUE]c[RVSON,C
    7]c[RVSOFF,BLUE]c[RVSON,C7]c
    [RVSOFF,BLUE]ccc[RVSON,C7]c[
    RVSOFF,BLUE]ccc[RVSON,C7]c[R
    VSOFF,BLUE]cc[RVSON,C7]c[RVSO
    FF,BLUE]ccc";
B6  510 PRINT"[RVSON,C7]cc[RVSOF
    F,BLUE]ccc[RVSON,C7]cc[RVSOF
    F,BLUE]c[RVSON,C7]c[RVSOFF,B
    LUE]c[RVSON,C7]c[RVSOFF,BLUE
    ]c[RVSON,C7]c[RVSOFF,BLUE]c[
    RVSON,C7]c[RVSOFF,BLUE]c[RVS
    ON,C7]c[RVSOFF,BLUE]c[RVSON,
    C7]c[RVSOFF,BLUE]c[RVSON,C7]
    c[RVSOFF,BLUE]c[RVSON,C7]cc[
    RVSOFF,BLUE]cc[RVSON,C7]ccc[
    RVSOFF]";
13  520 PRINT"[BLUE]c[RVSON,C7]c
    cc[RVSOFF,BLUE]c[RVSON,C7]c[
    RVSOFF,BLUE]c[RVSON,C7]c[RVS
    OFF,BLUE]cccccccccccccccccc
    cccccccccccccccccccccccccccc
    cccc";
C8  530 PRINT"cccccccccccccccccc
    cccccccccccccccccccccccccccc
    ccccccccccccccccccccccccccc";
14  540 PRINT"cccccccccccccccccc
    cccccccccccccccccccccccccccc
    cccccc[WHITE]IN[RIGHT]OPERAT
    ION...[BLUE]ccc";
2C  550 PRINT"cccccccccccccccccc
    cccccccccccccccccccccccccccc
    ccccccccccccccccccccccccccc";
BA  560 PRINT"cccccccccccccccccc
    cccccccccccccccccccccccccccc
    ccccccccccccccccccccccccccc";
F6  570 PRINT"[SC25,RIGHT,CYAN,C
    *,RVSON,SMS]TRACK:[SPC3]SECT
    OR:[SPC3]CODE:  [SNS,RVSOFF,
    S\,HOME]"
AB  580 POKE53265,27:RETURN
36  590 T$=STR$(T):B$=STR$(B)
04  600 T$=RIGHT$(T$,LEN(T$)-1)
94  610 B$=RIGHT$(B$,LEN(B$)-1)
47  620 IFLEN(T$)=1THENT$="0"+T$
BF  630 IFLEN(B$)=1THENB$="0"+B$
B5  640 PRINT"[RVSON,HOME,DOWN24
    ,RIGHT13]"T$"[RIGHT8]"B$"[RI
    GHT6]"CD$"[HOME]"
02  650 RETURN
9A  660 GOSUB890
58  670 FORT=1TO35:B=0
83  680 GOSUB930
E0  690 B=B+1:IFB<T(T)+1THEN680
FC  700 NEXT:RUN
69  710 INPUT"TRACK";T:GOSUB890
01  720 FORB=0TOT(T):GOSUB930
12  730 NEXT:RUN
96  740 INPUT"FROM TRACK";T1
F4  750 INPUT"TO TRACK";T2:GOSUB
    890
F8  760 FORT=T1TOT2:B=0
22  770 GOSUB930
67  780 B=B+1:IFB<T(T)+1THEN770
5F  790 NEXT:RUN
34  800 INPUT"TRACK";T
C2  810 INPUT"BLOCK";B:GOSUB890
F0  820 GOSUB930
CB  830 RUN
3C  840 INPUT"TRACK";T
D7  850 INPUT"FROM BLOCK";B1
DF  860 INPUT"TO BLOCK";B2:GOSUB
    890
AE  870 FORB=B1TOB2:GOSUB930
A1  880 NEXT:RUN
13  890 PRINT"[DOWN]INSERT DISC
    TO BE "A$"SCRAMBLED"
FF  900 PRINT"AND PRESS ANY KEY.
    .."
60  910 POKE198,0:WAIT198,1:POKE
    198,0
AD  920 GOTO380
94  930 GOSUB590
8E  940 OPEN15,8,15:OPEN5,8,5,"#
    "
59  950 PRINT#15,"U1:"5;0;T;B
B3  960 PRINT#15,"B-P:"5;3
30  970 FORA=3TO255:GET#5,A$
8E  980 POKE49152+A,ASC(A$+CHR$(
    0))
E9  990 NEXT:PRINT#15,"B-P:"5;3
80  1000 IFNSTHENSYS679
1A  1010 FORA=3TO255
DE  1020 PRINT#5,CHR$(PEEK(49152
    +A));:NEXT
86  1030 PRINT#15,"U2:"5;0;T;B:C
    LOSE5
84  1040 CLOSE15:RETURN
27  1050 DATA20,20,20,20,20,20,2
    0,20,20,20,20,20,20,20,20,20
    ,20,18,18,18,18,18,18
48  1060 DATA18,17,17,17,17,17,1
    7,16,16,16,16,16,160,3,185,0
    ,192,69,252,153,0,192
29  1070 DATA24,165,251,101,252,
    133,253,165,252,133,251,165,
    253,133,252,200,192,0
4C  1080 DATA208,228,96
```

Machine Code Disk Programming

Continuing his series on disk drives, Fergal Moane unravels the mysteries of Machine Code

If you can handle your disk drive in Basic, Machine Code programming comes very naturally. It demands slightly more work, but the Kernal and DOS still do nearly all the work for you. Machine code disk commands have a close link with Basic – they both use the Kernal, Commodore's I/O routines. The difference is that machine code routines, especially file handling, happen at lightning speed. This article will concentrate on file handling, the routines, and practical examples.

The Kernal

By the way, that's Commodore's spelling, not mine! Note that all the below routines are called with the JSR instruction, with the appropriate registers conveying data. Below is a flowchart for the use of the routines:

```
            SETLFS
               •
            SETNAM
               •
             OPEN
           •       •
input   •             •   output
CHKIN                     CHKOUT
  •                          •
CHRIN                     CHROUT
  •                          •
      •                 •
            CLOSE
               •
            CLRCHN
```

Here is a summary of the necessary routines:

SETLFS $FFBA

A FILE NUMBER
X DEVICE NUMBER
Y SECONDARY ADDRESS

This sets up parameters for use with any disk I/O, and is equivalent to the first three numbers in an OPEN statement.

SETNAM $FFBD

A NAME LENGTH
X LOWBYTE OF START OF NAME
Y HIGH BYTE OF START OF NAME

Sets a name for disk I/O. Note that for disks, a name must always be specified, except opening a to channel 15 (OPEN 15,8,15)

OPEN $FFCO

NO PARAMETERS REQUIRED

Use to open a file after SETLFS and SETNAM

CLOSE $FFC3

A FILE NUMBER

Closes the specified file

CHKIN $FFC6

X FILE NUMBER

Sets up a channel for input, after using the OPEN command

CHKOUT $FFC9

X FILE NUMBER

Sets up a channel for output, after using the OPEN command

CHRIN $FFCF

A DATA INPUT

Inputs data from the input channel defined by CHKIN, storing it in the accumulator. Equivalent to GET

CHROUT $FFD2

A DATA OUTPUT

Outputs the data in the accumulator to the output channel defined in CHKOUT. Equivalent to PRINT

CLRCHN $FFCC

NO PARAMETERS REQUIRED

Returns all input to the keyboard, and output to the screen. Use after finishing your own I/O

Examples

Here are two assembly listings to demonstrate the use of the above routines.

```
                 ;
                 ;READING DISK FILES
                 ;
SETLFS          =$FFBA
SETNAM          =$FFBD
OPEN            =$FFC0
CLOSE           =$FFC3
CHKIN           =$FFC6
CHRIN           =$FFCF
CHROUT          =$FFD2
CLRCHN          =$FFCC
                 ;
                 ;
                 *=$C000
                 ;
                 ;
INIT             LDA #$09           ;NAME LENGTH 9
                 LDX #NAME&255      ;LOW/HIGH BYTE NAME
                 LDY #NAME/256
                 JSR SETNAM
                 LDA #$01           ;FILE NUMBER
                 LDX #$08           ;DEVICE NUMBER
                 LDY #$05           ;CHANNEL NUMBER
                 JSR SETLFS
                 JSR OPEN           ;OPENS FILE
                 LDX #$01           ;FILE 1 AS INPUT
                 JSR CHKIN
LOOP             JSR CHRIN          ;INPUT IN A
                 JSR CHROUT         ;OUTPUT TO SCREEN FROM A
                 LDY $90            ;STATUS VARIABLE
                 CPY #$00           ;END OF FILE?
                 BEQ LOOP           ;NO. GET MORE DATA
END              LDA #$01           ;CLOSE FILE 1
                 JSR CLOSE
                 JSR CLRCHN         ;RETURN TO NORMAL
                 RTS                ;TO BASIC OR CALLING ROUTINE
NAME             .BYTE  'TEST FILE'
```

```
                 ;
                 ;DISK ERROR READ
                 ;
SETLFS          =$FFBA
SETNAM          =$FFBD
OPEN            =$FFC0
CLOSE           =$FFC3
CHKIN           =$FFC6
CHRIN           =$FFCF
CHROUT          =$FFD2
CLRCHN          =$FFCC
                 ;
                 ;
                 *=$C000
                 ;
                 ;
                 LDA #$0F
                 LDX #$08
                 LDY #$0F           ;OPEN 15,8,15
                 JSR SETLFS
                 LDA #$00
                 JSR SETNAM         ;NAME AND LENGTH IRRELEVANT
                 JSR OPEN           ;OPEN FILE
                 LDX #$0F
                 JSR CHKIN          ;INPUT CHANNEL
LOOP             JSR CHRIN          ;GET BYTE
                 CMP #$0D           ;EXIT IF RETURN
                 BEQ EXIT
                 JSR CHROUT         ;OUTPUT TO SCREEN
                 BNE LOOP           ;GET MORE DATA
EXIT             LDA #$0F
                 JSR CLOSE          ;CLOSE 15
                 JSR CLRCHN         ;NORMAL I/O
                 RTS                ;RETURN TO BASIC
```

For more information on KERNAL routines, see the Programmer's Reference Guide. Next time, I will present a complete DOS which demonstrates the above routines in a practical situation.

OOPS!

Unfortunately, we left four listings out from of June's installment of 'Machine Code Disk Programming', so we've reproduced them below.

DISK NAME

```
10 REM*****************************
********
20 REM*
      *
30 REM*          CHANGE DISK NAME
      *
40 REM*
      *
50 REM*****************************
********
60 PRINT"■CHANGE DISK NAME":PRIN
T"----------------"
70 PRINT"■ENTER NEW NAME OF DISK
"
80 INPUT"■";X$
90 REM******************************
*******
100 REM* OPEN TWO CHANNELS
110 REM****************************
*******
120 OPEN15,8,15:OPEN5,8,5,"#"
130 REM****************************
*******
140 REM POSITION IN DIRECTORY TR
ACK
150 REM****************************
*******
160 PRINT#15,"U1:";5;0;18;0
170 PRINT#15,"B-P:";5;144
180 PRINT"■■CURRENT DISK NAME IS
: ";
190 FORX=1TO16:GET#5,A$:IFA$=CHR
$(160)THEN210
200 PRINTA$;:NEXT
210 PRINT:PRINT"NEW DISK NAME IS
: ";X$
220 PRINT#15,"B-P:";5;144
230 REM****************************
*******
240 REM PAD OUT WITH SHIFTED SPA
CES
250 REM****************************
*******
260 IFLEN(X$)<16THENX$=X$+CHR$(1
60):GOTO260
270 REM****************************
*******
280 REM WRITE TO BUFFER AND CHAN
GE NAME
290 REM****************************
*******
300 PRINT#5,X$;
310 PRINT#15,"U2:";5;0;18;0
320 REM****************************
*******
330 REM INITIALIZE DISK AND DIRE
CTORY
340 REM****************************
*******
350 PRINT#15,"I":CLOSE5:CLOSE15
360 PRINT"■LOAD"+CHR$(34)+"$"+CH
R$(34)+",8":PRINT"■■■■LIST"
370 POKE198,3:POKE631,19:POKE632
,13:POKE633,13
380 NEW
```

UNSCRATCH

```
10 REM*****************************
********
20 REM*
      *
30 REM*          UNSCRATCH FILES
      *
40 REM*
      *
50 REM*****************************
********
60 PRINT"■UNSCRATCH FILES":PRINT
"---------------"
70 PRINT "■ENTER SCRATCHED FILE-
NAME"
80 PRINT"■ENTER END TO EXIT"
90 INPUTN$: IF N$="END" THEN 640
100 T=18:S=1:W=0
110 REM****************************
*******
120 REM PAD OUT WITH SHIFTED SPA
CES
130 REM****************************
*******
140 IFLEN(N$)<16THENN$=N$+CHR$(1
60):GOTO140
150 REM****************************
*******
160 REM OPEN FILES & ERROR CHECK
170 REM****************************
*******
180 OPEN 1,8,15,"I":GOSUB 580
190 OPEN 2,8,2,"#":GOSUB 580
200 REM****************************
*******
210 REM SEARCH TRACH 18 FOR NAME
220 REM****************************
*******
230 PRINT#1,"U1:";2;0;T;S:GOSUB
580
240 PRINT#1,"B-P:";2;0:GET#2,A$,
B$
250 TR=ASC(A$+CHR$(0)):SC=ASC(B$
+CHR$(0)):H=2
260 PRINT#1,"B-P:";2;H:GET#2,T$
270 C=ASC(T$+CHR$(0)):IFC>0THEN
370
280 GET#2,A$,B$:F$="":FOR X=1 TO
 16
290 GET#2,T$:F$=F$+T$:NEXT:IFF$=
""THEN390
300 REM****************************
*******
310 REM CHECK IF NAMES MATCH
320 REM****************************
*******
330 IFF$=N$THENGOTO410
340 REM****************************
*******
350 REM NEXT DIRECTORY ENTRY
360 REM****************************
*******
370 H=H+32:IFH<256THEN260
380 IFTR>0THENT=TR:S=SC:GOTO 230
390 IFWTHEN PRINT"■ END OF DIREC
TORY":GOTO 540
400 PRINT "■";N$;" NOT FOUND IN
DIRECTORY":CLOSE1:CLOSE2:END
410 PRINT"■";F$;" FOUND"
420 PRINT"■UNSCRATCH FILE (Y/N)"
:PRINT
430 INPUT X$: IF X$="Y" THEN 450
440 IF W THEN 370
450 PRINT "■1-SEQ  2-PRG  3-USR
 4-REL"
460 PRINT "ENTER FILE-TYPE (1-4)
 ? 2■■■";
```

```
470 INPUT P: IF P<1 OR P>4 THEN
PRINT "□□": GOTO 440
480 REM**************************
*******
490 REM WRITE FILE TYPE TO DISK
500 REM**************************
*******
510 PRINT#1,"B-P:";2;H: PRINT#2,
CHR$(P+128);
520 PRINT#1,"U2:";2;0;T;S:GOSUB
580:G=1
530 PRINT "⊠";F$;" UNSCRATCHED":
IF W THEN 370
540 GOSUB580:GOTO640
550 REM**************************
*******
560 REM DISK ERROR CHECK
570 REM**************************
*******
580 INPUT#1,E,M$,L,M:IF E=0 THEN
 RETURN
590 PRINT "⊠ ERROR: ";E;M$;L;M
600 REM**************************
*******
610 REM VALIDATE DISK TO RESTORE
 BAM ON              DISK
620 REM OTHERWISE, COPY TO ANOTH
ER DISK
630 REM**************************
*******
640 PRINT"⊠VALIDATE DISK (Y/N) ?
"
650 GETX$:IFX$=""THEN650
660 IFX$="N"THENCLOSE1:CLOSE2:EN
D
670 PRINT"□VALIDATING DISK TO UP
DATE BAM"
680 PRINT"⊠DO NOT REMOVE DISK UN
TIL LIGHT IS OUT"
690 OPEN15,8,15,"V":END
```

LOAD ADDRESS

```
10 REM***************************
********
20 REM*
      *
30 REM*        CHANGE LOAD ADDRES
S      *
40 REM*
      *
50 REM***************************
********
60 PRINT"□CHANGE LOAD ADDRESS":P
RINT"────────────────────"
70 PRINT "⊠ENTER PROGRAM-NAME"
80 PRINT"⊠ENTER END TO EXIT":PRI
NT
90 INPUTN$:IF N$="END" THEN 770
100 OPEN 1,8,15,"I":GOSUB 810
110 OPEN 2,8,2,"0:"+N$+",P,R"
120 REM**************************
*******
130 REM FILES OPEN, SO CHECK FOR
 ERRORS
140 REM**************************
*******
150 INPUT#1,E,M$,J,K: IF E=0 THE
N 240
160 CLOSE 1: CLOSE 2: PRINT "⊠ F
ILE:   ";N$;
170 IF E=62 THEN PRINT " NOT FOU
ND": GOTO 200
180 IF E=64 THEN PRINT " NOT A P
ROGRAM": GOTO 200
190 PRINT "⊠ DISK READ-ERROR";E:
 END
200 FOR J=1 TO 2000: NEXT: GOTO
60
210 REM**************************
*******
220 REM GET LOAD ADDRESS
230 REM**************************
*******
240 GET#2,A$,B$: IF ST THEN 190
250 CLOSE 2:CLOSE 1
260 A=ASC(A$+CHR$(0)):B=ASC(B$+C
HR$(0))
270 PRINT "⊠CURRENT LOAD-ADDRESS
 IS:"
280 PRINT "⊠";B*256+A
290 PRINT "⊠NEW LOAD-ADDRRESS IN
 DECIMAL ? 2049   ■■■■■■■■■■";
300 INPUT P: IF P<0 OR P>65535 T
HEN PRINT "□□": GOTO 290
310 REM**************************
*******
320 REM CALCULATE LOW/HIGH BYTE
AND PUT            INTO STRING FOR
WRITE COMMAND
330 REM**************************
*******
340 B=INT(P/256):A=P-B*256:P$=CH
R$(A)+CHR$(B):PRINT"⊠LOAD ADDRES
S: "P
350 PRINT "⊠WRITE THIS LOAD-ADDR
ESS (Y/N) ? Y■■■";
360 INPUT X$: IF X$<>"Y" THEN 70
370 REM**************************
*******
380 REM PAD OUT WITH SHIFTED SPA
CES
390 REM**************************
*******
400 J=LEN(N$):IFJ=16THEN420
410 FOR X=J+1 TO 16:N$=N$+CHR$(1
60):NEXT
420 OPEN 1,8,15: OPEN 2,8,2,"#"
430 GOSUB 810: T=18: S=1
440 PRINT#1,"U1:";2;0;T;S: GOSUB
 810
450 PRINT#1,"B-P:";2;0: GET#2,A$
,B$
460 REM**************************
*******
470 REM FIND TRACK AND SECTOR OF
 PRG
480 REM**************************
*******
490 T=ASC(A$+CHR$(0)):S=ASC(B$+C
HR$(0)):H=2
500 PRINT#1,"B-P:";2;H: GET#2,T$
510 C=ASC(T$+CHR$(0)):IFC<>130TH
EN580
520 REM**************************
*******
530 REM SEARCH FOR PROGRAM NAME
540 REM**************************
*******
550 GET#2,A$,B$:F$="":FOR X=1 TO
 16
560 GET#2,T$: F$=F$+T$: NEXT
570 IF F$=N$ THEN 660
580 H=H+32:IFH<256THEN500
590 REM**************************
*******
600 REM ADD 32 TO POINTER FOR NE
XT                DIRECTORY ENTRY
610 REM**************************
*******
620 GOTO 440
630 REM**************************
*******
640 REM ADDING CHR$(0) IN CASE O
F NULL          VALUE
650 REM**************************
*******
660 A=ASC(A$+CHR$(0)):B=ASC(B$+C
HR$(0))
670 REM**************************
*******
680 REM NOW CHANGE LOAD ADDRESS
690 REM**************************
*******
700 PRINT#1,"U1:";2;0;A;B: GOSUB
 810
710 PRINT#1,"B-P:";2;2:PRINT#2,P
$;
720 PRINT#1,"U2:";2;0;A;B: GOSUB
 810
730 CLOSE 2: GOSUB 810: CLOSE 1
740 PRINT "⊠LOAD-ADDRESS CHANGED
 FOR ";N$
750 PRINT "⊠CHANGE ANOTHER PROGR
AM (Y/N) ? N■■■";
760 INPUT X$: IF X$="Y" THEN 60
770 PRINT "□";: END
780 REM**************************
*******
790 REM DISK ERROR CHECK
800 REM**************************
*******
810 INPUT#1,E,M$,J,K:IF E=0 THEN
 RETURN
820 PRINT "⊠ ERROR: ";E;M$;J;K
830 CLOSE 2: CLOSE 1: END
```

PROTECT FILE

```
10 REM***************************
********
20 REM*
      *
30 REM*        PROTECTING FILES
      *
40 REM*
      *
50 REM***************************
********
55 PRINT"□SOFTWARE PROTECTION":P
RINT"────────────────────"
60 PRINT"⊠⊠⊠PROTECT (1) FILES"
70 PRINT"■■■■■■■■⊠(2) DISK"
80 GETA$:IFA$=""THENGOTO80
90 IFA$<"1"ORA$>"2"THEN80
100 ONVAL(A$)GOTO110,360
101 REM**************************
*******
102 REM SCRATCH PROTECT FILES
103 REM**************************
*******
110 PRINT"□PROTECT FILES":PRINT"
────────────────────"
120 PRINT"⊠(1) PROTECT FILE  (2)
 UNPROTECT FILE"
130 GETA$:IFA$=""THENGOTO130
140 IFA$<"1"ORA$>"2"THEN130
150 IFA$="1"THENPT=64:REM FILE T
YPE CODE TO PROTECT FILES
160 IFA$="2"THENPT=0:REM FILE TY
PE CODE TO UNPROTECT FILES
170 PRINT"⊠ENTER FILE TO BE ";:I
FPT=0THENPRINT"UN";
175 PRINT"PROTECTED":PRINT"⊠ENTE
R END TO EXIT⊠"
180 INPUTN$:IFN$="END"THENGOTO50
185 REM**************************
*******
186 REM VERIFY FILE EXISTS
187 REM**************************
*******
190 OPEN15,8,15:OPEN1,8,5,"0:"+N
$+",R":GOSUB340
191 REM**************************
*******
192 REM PAD OUT NAME WITH SHIFTE
D SPACE
193 REM**************************
*******
200 IFLEN(N$)<16THENN$=N$+CHR$(1
60):GOTO200
201 REM**************************
*******
```

```
202 REM SEARCH DIRECTORY TRACK F
OR FILE
203 REM*************************
*******
210 CLOSE1:CLOSE15:OPEN15,8,15:O
PEN5,8,5,"#":T=18:S=1:GOSUB340
220 PRINT#15,"U1:";5;0;T;S:PRINT
#15,"B-P:";5;0:GET#5,A$,B$:Y=T:Z
=S
230 T=ASC(A$+CHR$(0)):S=ASC(B$+C
HR$(0)):H=2
240 PRINT#15,"B-P";5;H:GET#5,T$
250 C=ASC(T$+CHR$(0))
260 GET#5,A$,B$:F$="":FORX=1TO16
:GET#5,T$:F$=F$+T$:NEXT
270 PRINT"SEARCHING : ";F$:PRINT
"";:IFN$=F$THEN300
271 REM*************************
*******
272 REM NEXT TRACK AND SECTOR
273 REM*************************
*******
280 H=H+32:IFH<256THEN240
290 GOTO220
300 PRINT#15,"B-P";5;H
301 IFPT=OTHENC=C-64:REM C IS CU
RRENT FILE TYPE
302 REM*************************
*******
303 REM CHANGE ACTUAL FILE TYPE
304 REM*************************
*******
310 PRINT#5,CHR$((C+PT)OR128);:R
EM OR 128 CLOSES THE FILE PROPER
LY
320 PRINT#15,"U2:";5;0;Y;Z
330 CLOSE1:GOSUB340:CLOSE15
334 PRINT"■■":IFPT=OTHENPRINT"UN
";
335 PRINT"PROTECTED"
336 FORU=OTO1000:NEXT:GOTO50
337 REM*************************
*******
338 REM DISK ERROR CHECK
339 REM*************************
*******
340 INPUT#15,E,E$:IFE=OTHENRETUR
N
350 PRINT"■■DISK ERROR: ";E;E$:C
LOSE5:CLOSE15:END
355 REM*************************
*******
356 REM WRITE PROTECT DISK
357 REM*************************
*******
360 PRINT"■WRITE PROTECT DISK":P
RINT"──────────────────"
370 PRINT"■(1)PROTECT DISK  (2)U
NPROTECT DISK"
380 GETA$:IFA$=""THENGOTO380
390 IFA$<"1"ORA$>"2"THEN380
395 PRINT"■■CHECK DISK AND PRESS
 ANY KEY":POKE198,0:WAIT198,1:PO
KE198,0
400 ONVAL(A$)GOTO410,450
401 REM*************************
*******
402 REM PROTECT DISK
403 REM*************************
*******
410 OPEN15,8,15:OPEN5,8,5,"#":GO
SUB340
420 PRINT#15,"U1:";5;0;18;0:PRIN
T#15,"B-P:";5;2:PRINT#5,CHR$(66)
;
430 PRINT#15,"U2:";5;0;18;0
440 GOSUB340:PRINT#15,"I":CLOSE5
:CLOSE15:GOTO60
441 REM*************************
*******
442 REM UNPROTECT DISK
443 REM*************************
*******
450 OPEN15,8,15,"I":OPEN2,8,2,"#
"
460 PRINT#15,"U1:";2;0;18;0:PRIN
T#15,"B-P:";2;2:PRINT#2,CHR$(65)
;
470 PRINT#15,"B-P:";2;166:PRINT#
2,CHR$(65);
480 PRINT#15,"M-W";CHR$(1);CHR$(
1);CHR$(1);CHR$(65)
490 PRINT#15,"M-W";CHR$(2);CHR$(
7);CHR$(1);CHR$(65)
500 PRINT#15,"U2:";2;0;18;0
510 GOTO440
```

continued from page 32

```
    TERPOISE,COUNTERVAIL,OFFSET,
    *
62  3900 DATACOUNTERFEIT,FALSIFY
    ,COPY,SIMULATE,FAKE,IMITATE,
    PHONEY
AF  3910 DATARECORD,NOTE,COPY,DU
    PLICATE,FORGE,PASTICHE
B3  3920 DATASHAM,REPRODUCTION(A
    RT),*
F6  3930 DATAFUNERAL,BURIAL,INHU
    MATION,INTERMENT,*
7B  3940 DATAIMPOSSIBLE,HOPELESS
    ,FORLORNE,IMPRACTICABLE,DIFF
    ICULT,HARD(DIFFICULT)
13  3950 DATATRICKY,TRYING,*
03  3960 DATAVENOM,POISON,TOXIN,
    *
1B  3970 DATASURFACE,COVERING(FR
    ONT),EXTERIOR,VENEER,FACADE(
    FRONT),FRONTAGE,*
9F  3980 DATASTRONG,POWERFUL,GRE
    AT(STRONG),RESILIENT,FORTHRI
    GHT,DURABLE,*
C5  3990 DATAWOUND,INJURY,CUT(IN
    J),LESION,SCAR(INJ),SLASH,LA
    CERATE,*
A6  4000 DATALOSE,MISPLACE,MISLA
    Y,*
DD  4010 DATAMAGIC,ENCHANTMENT,O
    CCULT,SORCERY,SPELL,CONJURY,
    ILLUSION,TRICK,*
11  4020 DATAMAJESTIC,EXALTED,GR
    AND,IMPERIAL,KING,NOBLE,REGA
    L,ROYAL,STATELY,*
75  4030 DATABUREAUCRACY,ADMINIS
    TRATION,CIVIL SERVICE,DIRECT
    ORATE,GOVERNMENT
E8  4040 DATAMINISTRY,OFFICIALDO
    M,*
1F  4050 DATABUY,PURCHASE,AQUIRE
    ,OBTAIN,GET,SECURE,RENT,LEAS
    E,HIRE,*
C1  4060 DATAMINE,COAL FACE,COLL
    IERY,PIT,SHAFT,QUARRY,TUNNEL
    ,EXCAVATION,*
41  4070 DATAMINOR,INCONSEQUENTI
    AL,INSIGNIFICANT,INFERIOR,NE
    GLIGIBLE,PETTY
69  4080 DATASUBORDINATE,TRIFLIN
    G,TRIVIAL,UNIMPORTANT,*
E5  4090 DATAMAD,CRAZY,CRACKERS,
    DERANGED,INSANE,LUNATIC,DAFT
    ,STUPID,*
A7  4100 DATACUSTOMER,CONSUMER,P
    UNTER,CLIENT,BUYER,PATRON,SH
    OPPER,PURCHASER,*
F4  4110 DATAETCH,CARVE,CUT(CARV
    E),INSCRIBE,INCISE,*
B7  4120 DATAESSENTIAL,CRUCIAL,I
    MPORTANT,NEEDED,REQUISITE,VI
    TAL,INTRINSIC,*
EC  4130 DATAFLAVOUR,ESSENCE,TIN
    CTURE,ZEST,*
2D  4140 DATACRITIC,ICONOCLAST,D
    ISSIDENT,HERETIC,RADICAL,REB
    EL,*
0C  4150 DATAIDENTICAL,ALIKE,COR
    RESPONDING,DUPLICATE,EQUAL,E
    QUIVALENT
11  4160 DATALIKE,SAME,TWIN,MATC
    H(LIKE),*
D3  4170 DATAINTERESTING,ABSORBI
    NG,APPEALING,COMPELLING,ENGA
    GING,INTRIGUING,*
AE  4180 DATAEXTOL,ACCLAIM,APPLA
    UD,CELEBRATE,COMMEND,PRAISE,
    *
CA  4190 DATALIST,CATALOGUE,DIRE
    CTORY,FILE,INDEX,INVENTORY,I
    NVOICE,RECORD(LIST)
0F  4200 DATA**
```

Inside the

Continuing his series on programming the 1541 drive, Fergal Moane offers some insight into the internals of your machine

By Fergal Moane

Now that we've gained a fair knowledge of machine code and disk editing, it's time to turn our attention to the internal memory of the 1541. Inside the 1541 lies a dedicated computer. It has a 6502 processor (as in the 64), 16K ROM, 2K RAM, and interface chips.

Commodore's designers have for once had foresight, and given us a host of advanced commands to manipulate the drive's internal memory. You can use the drives ROM routines, or write your own in the RAM buffers. This allows advanced protection routines and disk turbos to be created. I've even seen a program which uses the drive's 6502 to calculate numbers for vector graphics routines, effectively using the drive as a second slave processor, speeding up things no end.

The Commands

The three commands below are the approximate equivalent of PEEK, POKE and SYS. Using these commands, it's possible to create your own machine code, or use the DOS routines. Unfortunately, Commodore has never published disassemblies of its disk ROM, so use of it is extremely limited. The best way to use the disk memory is to buy a machine code minotor which allows assembly and disassembly to disk memory. DATEL and TRILOGIC have cartridges with monitors possessing this facility.

MEMORY-READ
PRINT #15, "M-R" CHR$ (lo) CHR$ (hi) CHR$ (number)
CHRS (lo) is the low byte of the address in DOS that is to be read.
CHR$ (hi) is the high byte of the address.
CHR$ (number) is an optional parameter which indicates the number of bytes to be read.

This presumes that OPEN 15,8,15 has been performed earlier in the program (as it always should be). A GET #15 will read the byte from the error channel, peforming a PEEK of the address specified by the CHR$ codes.

MEMORY-WRITE
PRINT # 15, "M-R" CHR$ (lo) CHR$ (hi) CHR$ (number) CHR$ (data) CHR$ (data) etc...

Number obviously specifies the number of bytes to be written. The number of pieces of data should correspond to the number in the third CHR$ command. The maximum amount of data that can be sent at one time is 34 bytes.

MEMORY-EXECUTE
PRINT # 15, "M-E" CHR$ (lo) CHR$ (hi)

This command will call and execute a machine code program that resides in the DOS memory. This routine should end with a RTS instruction. DOS routines can be used with this command.

USER COMMANDS
PRINT # 15, "Un"

See the table below for possible values of N. These commands allow a jump table jo be set up, as there is enough room for a JMP instruction to your routine. This means that by using two letters, you can access a table of routines quickly and easily, even from Basic. All the below User commands access buffer 2, the unused buffer suitable for machine code.

U3 or UC	jump to $0500
U4 or UD	jump to $0503
U5 or UE	jump to $0506
U6 or UF	jump to $0509
U7 or UG	jump to $050C
U8 or UH	jump to $050F
U9 or UI	jump to SFFFA
U; or UJ	power-up vector, resets drive

Examples

Here are some simple examples which demonstrate the above theory. They may be useful in disk utility programs.

EXAMPLE 1 - space on the current disk

```
10 OPEN15, 8, 15, "I"
20 PRINT # 15, "M-R" CHR$ (250) CHR$ (2)
30 GET #15, X$:X$ = X$ + CHR$(0)
40 PRINT  15, "M-R" CHR$ (252) CHR$ (2)
50 GET #15, Y$:Y$ =Y$ + CHR$(0)
60 A=ASC (X$) + 256* ASC (Y$)
70 PRINTA "BLOCKS FREE
80 PRINTA/4 "KILOBYTES FREE": PRINT (A/4)* 1024 "BYTES FREE"
90 CLOSE 15
```

This reads the blocks free on a disk, and calculates the number of kilobytes available by dividing by four. X$ and

e 1541

Y$ contain the low and high bytes of the blocks free.

EXAMPLE 2 -alignment adjuster and woodpecker remover

```
10 INPUT "WOODPECKER (Y/N)"
;W$
20 IFW$ < >"Y" THEN B =B + 128
30 INPUT"HALF TRACK SEEKER
(Y/N)" ;A$
40 IFA$ < >"Y" THEN B=B+64
50 INPUT "LOADING ATTEMPTS
(0-30)" ;L
60 B=B+L: OPEN 15, 8, 15
70 PRINT # 15, "M-W" CHR$ (106)
CHR$ (0) CHR$ (1) CHR$ (B)
80 CLOSE15
```

This program alters the vital location 106 in disk RAM. It controls the number of read attempts to be made, i.e. how many times the disk drive will try to read the sector. This is usually five. The 'woodpecker' sound made when the disk head hammers off the end stop can be avoided. Also, if your disk drive is out of alignment, the half track reads will increase the chance of loading a misaligned program.

EXAMPLE 3 - disk name

```
10 OPEN 15, 8, 15, "I"
20 PRINT # 15, "M-R" CHR$ (144)
CHR$ (7) CHR$ (16)
30 INPUT # 15, NAME$
40 PRINTNAME$: CLOSE 15
```

This reads the sixteen letter name of the current disk. Note the CHR$(16) to define the number of locations to be read and the use of INPUT # to fetch multiple characters.

When fiddling around with your disk drive, it makes sense to take the disk out of the drive. This ensures that if a duff value hits a sensitive spot, your disk will not suffer the consequences. Remember that you cannot damage the drive by software, and switching it off and on again will restore everything to normal.

Memory Map

There's little point in messing around with drive memory if you don't know what you're doing. Below is a 1541 memory map which details some of the more interesting features I came across. I saw no point in providing DOS disassemblies, so you'll need a good disk monitor to experiment with DOS routines.

RAM 0000-07FF 0000-2047
ROM 0800-FFFF 2048-65535

HEX	DECIMAL	DESCRIPTION
0000	0	COMMAND CODE FOR BUFFER 0
0001	1	COMMAND CODE FOR BUFFER 1
0002	2	COMMAND CODE FOR BUFFER 2
0003	3	COMMAND CODE FOR BUFFER 3
0004	4	COMMAND CODE FOR BUFFER 4
0006-0007	6-7	TRACK AND SECTOR FOR BUFFER 0
0008-0009	8-9	TRACK AND SECTOR FOR BUFFER 1
000A-000B	10-11	TRACK AND SECTOR FOR BUFFER 2
000C-000D	12-13	TRACK AND SECTOR FOR BUFFER 3
000E-000F	14-15	TRACK AND SECTOR FOR BUFFER 4
0012-0013	18-19	ID FOR DRIVE 0
0014-0015	20-21	ID FOR DRIVE 1
0016-0017	22-23	CURRENT ID
0020-0021	32-33	HEAD TRANSPORT FLAG
0030-0031	48-49	BUFFER POINTER
0039	57	8-MARK FOR BEGINNING OF DATA BLOCK HEADER
003A	58	PARITY FOR DATA BUFFER
003D	61	DRIVE NUMBER FOR DISK CONTROLLER
003F	63	BUFFER NUMBER FOR DISK CONTROLLER
0043	67	NUMBER OF SECTORS PER TRACK
0047	71	7-MARK FOR BEGINNING OF DATA BLOCK HEADER
0049	73	STACK POINTER
004A	74	STEP COUNTER FOR HEAD TRANSPORT
0051	81	ACTUAL TRACK NUMBER FOR FORMATTING
0069	105	10-STEP SIZE FOR SECTOR DIVISION
006A	106	5-NUMBER OF READ ATTEMPTS
006F-0070	111-112	POINTER TO ADDRESS FOR M COMMANDS
0077	119	DEVICE NUMBER+32 FOR LISTEN
0078	120	DEVICE NUMBER+64 FOR TALK
0079	121	FLAG FOR LISTEN
007A	122	FLAG FOR TALK
007C	124	FLAG FOR ATN FROM SERIAL BUS
007D	125	FLAG FOR EOI FROM SERIAL BUS
007F	127	0-DRIVE NUMBER
0080	128	CURRENT TRACK NUMBER
0081	129	CURRENT SECTOR NUMBER
0082	130	CURRENT CHANNEL NUMBER
0083	131	CURRENT FILE NUMBER
0084	132	CURRENT SECONDARY ADDRESS
0085	133	CURRENT DATA BYTE
0088-008D	139-141	DIVISION WORK AREA
0094-0095	148-149	BUFFER POINTER
0099-009A	153-154	ADDRESS OF BUFFER 0
009B-009C	155-156	ADDRESS OF BUFFER 1
009D-009E	157-158	ADDRESS OF BUFFER 2
009F-00A0	159-160	ADDRESS OF BUFFER 3
00A1-00A2	161-162	ADDRESS OF BUFFER 4
00A3-00A4	163-164	POINTER TO INPUT BUFFER
00A5-00A6	165-166	POINTER TO ERROR MESSAGE BUFFER
00B5-00BA	181-186	RECORD NO. LO. BLOCK NO. LO
00BB-00C0	187-192	RECORD NO. HI. BLOCK NO. HI
00C1-00C6	193-198	WRITE POINTER FOR REL FILE
00C7-000C	199-204	RECORD LENGTH FOR REL FIKE
00D4	212	POINTER IN RECORD FOR REL FILE
00D5	213	SIDE SECTOR NUMBER
00D6	214	POINTER TO DATA BLOCK IN SIDE SECTOR
00D7	215	POINTER TO RECORD IN REL FILE
00E7	231	FILE TYPE
00F9	249	BUFFER NUMBER
0100-0145	256-325	STACK
0200-0228	512-552	INPUT BUFFER FOR COMMANDS
024A	586	FILE TYPE
025B	600	RECORD LENGTH
0259	601	TRACK-SIDE SECTOR
025A	602	SECTOR-SIDE SECTOR
0274	628	LENGTH OF INPUT LINE
0278	632	NUMBER OF FILENAMES
0297	663	FILE CONTROL METHOD
0280-0284	640-644	TRACK OF A FILE
0285-0289	645-649	SECTOR OF A FILE
02D5-02F9	725-761	BUFFER FOR ERROR MESSAGES
02FA-02FC	762-764	NUMBER OF BLOCKS FREE
0300-03FF	768-1023	BUFFER 0 -MAIN WORK BUFFER
0400-04FF	1024-1279	BUFFER 1 -DIRECTORY
0500-05FF	1280-1535	BUFFER 2 -USER BUFFER
0600-06FF	1536-1791	BUFFER 3 -DIRECTORY
0700-07FF	1792-2047	BUFFER 4- BAM IMAGE
0800-17FF	2048-6143	UNUSED ROM
1800-180F	6144-6159	SERIAL BUS CONTROLLER 6522
1810-1BFF	6160-7167	UNUSED ROM
1C00-1C0F	7168-7183	DRIVE CONTROLLER 6522
1C10-C0FF	7184-49407	UNUSED ROM
C100-FFFF	49408-65535	DISK OPERATING SYSTEM

Memory Man

Explore the intricasies of your 128's memory

By David Kelsey

The C128 has many programmable chips within it. There's the VICII chip for 40-column screen output, the SID chip, which is dedicated to sound, and the two CIA chips which control interfacing and interrupts on the C128. These chips are identical to those found on the C64 – except for the VICII chip, which has one extra register to control the clock speed of the 8502 processor – and has already been covered in a vast amount of detail. But the C128 has two extra chips. The first is the 80-column screen chip, the second is the Memory Mangement Unit (MMU for short). In this article, we'll be taking a closer look at the MMU.

C128 Memory

The C64 was very unique of its time simply because the micro processor could access more than 64K of memory, though not all at the same time. It could do this because the 6502 was upgraded to allow for special methods of 'bank' selection. RAM and ROM were separated into blocks, and parts of RAM could be mixed with parts of ROM to produce a full 64K of memory, which the 6510 microprocessor would then see as a full 64K of addressable memory.

The C128 also uses this concept, but Commodore decided to add another 64K of RAM and a whole lot more ROM. The 8502 (which is an upgraded 6510 to allow 2 mhz operating) couldn't cope with this amount of RAM and ROM, so a chip dedicated to managing all this memory was required. Thus the MMU was born. The upshot of this is that the Commodore has 128K of accessible RAM and a vast amount of ROM to provide the operating system, Basic V7.0, and of course the C64 operating system with Basic V2.0. All this is managed by the MMU so that a different variety of mixtures of RAM and ROM can be produced to make up the full 64K for the 8502 to address. Each one of these varieties is called a **configuration.**

Who uses the MMU

This chip is used frequently by the operating system. When running a Basic program, it has to retrieve the actual program from RAM, but the code for Basic which operates on the Basic instruction is stored in ROM.

You may have also used the MMU. The Basic command BANK provides control over the MMU. This command is only used to allow selection of the possible configuration to allow for running machine code programs from Basic or to poke/peek certain memory locations.

The 8502 registers

Before I start on the MMU registers, I'll briefly mention the 8502 registers used to manipulate what the 8502 addressing sees in the way of ROM and RAM. These registers are found at locations $00 and $01 – location $00 is the data direction register for register $01.

In 64 mode, they operate as standard to a C64, and much has already been written on this subject. In 128 mode, however, there is a difference. Bits 0-2 of register $01 are used to tell the VIC chip and the 8502 where to get information.

Bits 0 & 1

The VIC chip gets the colour information for the screen from RAM at $0800, which is part of the I/O block. However, there is another block of RAM which can be used for colour. This means there are two colour RAM blocks. They both reside at locations $d800 – $dfff, and a block is selected using bits 0 and 1 of address $01.

Bit 1 tells the VIC chip which block to use to display the colour – '0' represents RAM block 0, and a '1' represents RAM block 1. Bit 0 tells the MMU which block will be seen by the 8502 when the I/O section is available to the 8502 (more about this later). This allows the updating of one colour block seen by the 8502 while the other colour RAM block is actually being displayed by the VIC chip. It would then be possible to switch the blocks displayed and update the other RAM block.

On power up, the standard block displayed by VIC and seen by the 8502 is ram block 0.

Bit 2

Bit 2 selects whether the Character information is within the VIC video bank – 1 means that the character information is found at RAM within the video bank, while 0 means that the character information is taken from ROM at $D000.

More information can be found with the text regarding the VIC chip.

The MMU registers

The MMU is controlled using several registers which allow a programmer to control which blocks of RAM and ROM are concurrently visible to the 8502, select which micro processor is being used (the C128 has a Z80 inside), and a lot more besides.

What follows is a description of each register and its relative use.

The configuration register

The configuration register is the one that tell the MMU how to make up the 64K to be addressed by the 8502. This register is based at address $0500 in the I/O memory block, but can also be found at $FF00. The reason for this will become clear very soon.

The first thing this register can do is select which bank of RAM will be used – RAM 0-1 (0-3 is possible with memory expansion). The area from $C000-$FFFF can be varied in four different ways. It could contain kernal ROM, other types of ROM or just RAM. The area $8000-$bfff is also variable in a similar sort of fashion.

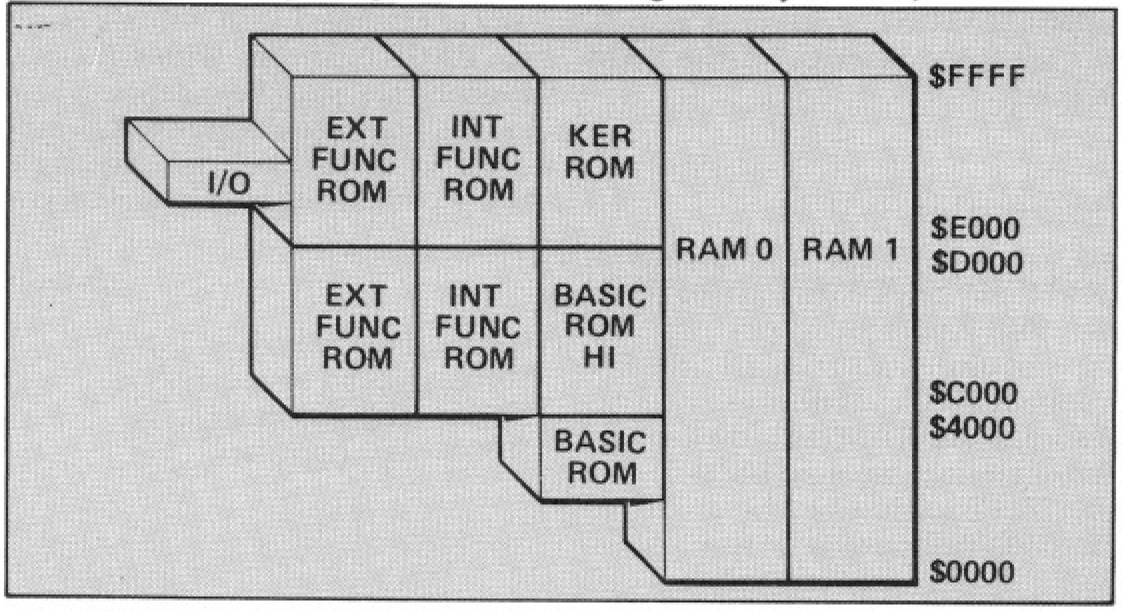
agement

Area $4000-$7fff can either be Basic ROM or RAM.

Lastly, the area £d000-$dfff can either allow or contain the I/O register or RAM. This is why $ff00 shadows the MMU register at $d500. It's possible to switch out the I/O area, but if you did that, how could you then change the 64K configuration again?

Figure 1 shows graphically how the RAM and ROM sit:

Figure 1



The following table summarises how a 64K block can be put together:

NAME OF AREA	BITS	SELECTION
Bank Select	7,6	00 = RAM 0
		01 = RAM 1
		10 = RAM 2
		11 = RAM 3
$c000-$ffff	5,4	00 = Kernal ROM
		01 = Internal ROM
		10 = External ROM
		11 = RAM
$8000-$bfff	3,2	00 =BASIC ROM high
		01 = Internal ROM
		10 = External ROM
		11 = RAM
$4000-$7fff	1	0 =BASIC ROM low
		1 = RAM
$d000-$dfff	0	0 = I/O registers
	1	1 = RAM/ROM (depending on $c000-$cfff)

For example, if you wanted access purely to RAM zero, you'd select bit pattern 00111111 = 3f and store this at location $ff00. If you wanted RAM 1 and kernal ROM with I/O registers, the bit pattern required would be 01001110 = 4E.

Preconfiguration Registers

The preconfiguration registers allow a way of setting up fixed configurations. You can have upto four fixed settings, and to do this you decide on the configuration you want, then store it in one of the preconfiguration registers $d501-$d504. Whenever you want that configuration, you just access the corresponding registers $ff01-$ff04. An access can be any type of store such as STA, STX or STY to that address. By way of example, consider this small program:-

```
LDA # $7F
STA $D501 (assume that I/O registers
are accessed)
.
.
.
.
STA  $FF01  (select configuration
  $7F)
```

The registers at $ff00-$ff04 must be available in both RAM blocks so as to allow configuration switching anywhere. A further useful point is that when an interrupt occurs, the system could be in any configuration, so at least part of the interrupt service routine must be available in all configurations. Instead of placing the code in 'COMMON RAM', the concept of which is described a little later, code is copied from ROM to RAM in both RAM blocks.

Mode configuration register

As you know the C128 has three modes of operation; C64, C128 and Z80 mode. Via this register these modes can be selected.

Below is a description of each bit of the register and its function. This register can be found at $d505, and isn't shadowed.

BIT **DESCRIPTION**

0 This selects the processor to be used. 0 = Z80, 1 = 8502.

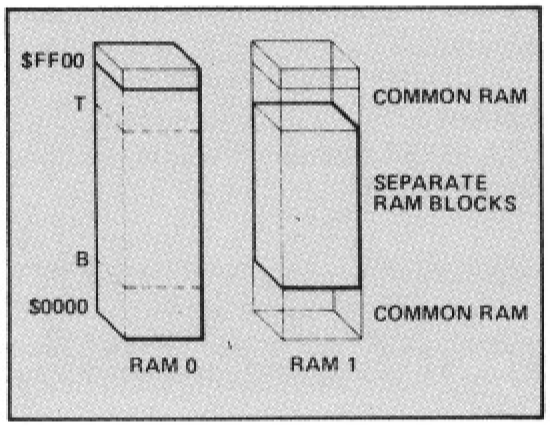
1,2 Not used.

3 This bit is called the FSDIR bit, and is used to control the flow of data along the CIA used for serial interfacing. An 0 indicates the fast serial port is for input, and a 1 means that the fast serial port is for output.

4,5 These are used for cartridge port. They can detect signals on the cartridge port and and act accordingly. For instance, a C64 cartridge pulls these lines low. The C128 detects this on power up and puts the computer into C64 mode to allow the cartridge software to run.

6 Selects C64 or C128 mode. 0 = C128 mode, 1 = C64.
7 40/80 display key sense. 1 = key depressed.

Figure 2.



Ram Configuration register

Although on a standard C128 there are two separate blocks of 64K RAM, it's possible to have 'Common' RAM. Common RAM is RAM which appears in RAM 0 and RAM 1. To illustrate this, try the following example:-

```
BANK 1:POKE 254,144
BANK 0:POKE 254,175
BANK 1:PRINT PEEK(254)
```

The result will be 175. This means that a certain part of memory is the same no matter which RAM block you are in. You change it while in RAM 0 and switch to RAM 1, but at that certain part of memory the 8502 still sees RAM 0. This register is the one that allows the programmer to control which parts of memory are common, and how much of it is common. This register is found at $d506.

Figure 2 gives a graphic illustration of the concept of 'common' RAM. From 'T' to $FFFF, the contents of the memory seen by the 8502 will always be the same no matter what ram block the 8502 is 'seeing'. Similarly, from $0000 to 'B' the same applies. Via this register you can move 'T' and 'B' around.

BITS	DESCRIPTION
1,0	Determine the amount of common Ram. 00 = 1K common RAM 01 = 4K common RAM 10 = 8K common RAM 11 = 16K common RAM
3,2	These determine which areas of memory will be common. It can be xK from the top of memory down, xK from the bottom of memory upwards, or both. xK is the amount specified to be common (explained above) 00 = No common RAM 01 = Bottom of RAM is common 10 = Top of RAM is common 11 = Both top and bottom are common
5,4	Not used.
6	VIC chip bank (see below).
7	Not used.

The VIC chip can be manipulated to use different parts of memory. This bit allows RAM block selection as well. 0 = use RAM block 0. 1 = Use RAM block 1.

Examples

4K common RAM with bottom of RAM only being common ($05):-
B = $0FFF (inclusive)
T = not used
common $0000 – $0fff
1K common RAM with the top and bottom being common ($0C):-
B = $03FF (inclusive)
T = $FC00 (inclusive)

i.e. common $0000 – $03ff, $fc00 – $ffff

Page 0 & 1 pointers

Another feature of the MMU is the ability to move page 0 and page 1 to any part of memory. The 8502 is transparent to this, and so the code isn't affected.

For example, suppose the MMU has 'moved' page zero to location $1000. Then, the example:-

```
LDA # $00
STA $60
LDA # $40
STA $61
LDY # 0
LDA ($60),Y
```

will actually put 00 40 into locations $1000 and $1001 respectively. The final line will of course execute as normal, placing in A the contents of address $4000.

Whenever the microprocessor accesses page 0, the MMU intecepts this and creates a new address based on the contents of the page 0 pointer. The microprocessor then uses this address thinking it's page zero. What this means is that you can move page zero to another free area if you've no spare memory in usual page zero. The microprocessor doesn't recognise any difference, and so all usual page zero 8502 instructions work. A similar description could be given for PAGE 1 pointer as it works in the same way.

Two registers are used to move the page 0 boundary, and two are required for page 1. The two sets of registers work in exactly the same way, so by way of example I shall just talk about the PAGE 0 pointer. Within the 64K memory are page boundaries. There are 256 page boundaries in 64K, which occur at every 256 bytes. The table below outlines the page boundaries:

$0000 – First boundary
$0100 – second boundary
$0200 – Third boundary
$0300 – next boundary

.

.

.

sfe00 – boundary
$ff00 – final boundary

The page 0 must be placed so that it starts at one of these boundaries. The address $D507 allows you to select 1 of the 256 boundaries you wist it to go on.

You can also select which block of RAM you want to place the pae you're moving. This is controlled by the register at $D508. Bits 0-3 are used to select the RAM block, but on a standard 128 the only bit that has any effect is bit 0. An '0' indicates RAM 0, and a '1' selects RAM block 1.

One further thing to note: to change the contents of these registers

requires a special operation. The registers are not changed until a write operation is performed on the lower register that controls page boundaries. For page 0 this would be the register at address $D507. You first write to the bank register at $D508 for page 0 (the value is stored but not in the register). When you write to the boundary register (at $D507 for page 0), this register is updated and then the bank register is updated with the value stored. For example, moving page 0 to location £1000 in bank 1 could be done as follows

```
LDA $01     set bank 1.
sta $D508   must be done before the
            $D508 write.
LDA $10     boundary for address $1000
            (ie high byte of address).
STA $D507
```

This isn't a very practical example – if you try this it will crash the computer because the operating system requires access to the correct page zero. A better demonstration is this short program:

```
SEI         stop ny interupts which
            may use the stack.
LDA $00
STA $D50A
LDA $04
STA $D509   set page 1 to address $0400.
LDA $EA     any character would do.
PHA
PHA
PHA
PHA
PHA
PHA
PHA
PHA
LDA $01
STA $d509   restore original stack.
CLI
BRK
```

When you run this, some characters will appear on the 40 column screen. Note that the stack position isn't preserved by this program, so if you run it again, characters will appear in different places.

To summarize the registers:-

$D507 – Boundary register for page 0.
$D508 – Bank register for page 0.
$D509 – Boundary register for page 1.
$D50A – Bank register for page 1.

MMU version Register

This register is found at address $D50B, and has two parts to it. The high nibble – i.e. bits 4-7 – contains the number of 64K RAM blocks that the MMU can 'see'. In the case of the standard C128, this is two blocks. The low nibble (bits 0-3) contains the version number of the MMU chip. The current value is 0, so for a standard C128, the contents of this register are $20.

Accessing memory blocks

OK, we've seen the registers that are available. The next thing to consider is how we can access memory currently not seen by the CPU, without appearing to have to change the configuration and run the risk of crashing because we switched out the memory the program was actually running in. This consideration is only required for machine code programmers. Basic contains the BANK command which, although not allowing all the configurations possible, does allow the ability to access any part of the range of commodore memory at any time.

The Commodore operating system provides five routines for just this situation. These are:-

LDAFAR
STAFAR
JSRFAR
JMPFAR
CMPFAR

One way to call these routines is via the kernal calls, which don't actually call these routines directly but go into kernal ROM first. I don't recommend this, as it restricts you to having the kernal routines visible to the CPU before you can access these routines. I found on one occasion, that I couldn't get access to KERNAL ROM but required these routines.

The first question you may ask yourself is how can these routines be available from any configuration? The answer lies in how Commodore initialises the MMU on startup. It defines a common area from $000 to $03ff, and places the routines within this block. Therefore, no matter which configuration you are in, these routines are always there and as only RAM is available at the address range $0000 – $3fff, these routines will always be available.

LDAFAR, STAFAR & CMPFAR

These routines provide extensions to the LDA ($xx),y STA ($xx),y and CMP ($xx),y 8502 commands. The format for calling is as follows:-

```
LDAFAR:  lda    xx      zero page address used 'xx' for LDAFAR
                        ($xx),y
         sta $02aa
         ldx config     config is the configuration required
         jsr $02a2
STAFAR:  ldx    xx
         stx $02b9
         ldx config
         jsr $02af
CMPFAR   ldx    xx
         stx $02c8
         ldx config
         jsr $02be
```

The kernal version is similar, but it allows the BANK number to be given rather than the configuration. It then converts the bank number to the required configuration, updates the routine to use the correct zero page address, then calls the LDAFAR or STAFAR routines. The call to these routines are as follows (making sure you have the correct configuration):-

```
LDAFAR:  lda    xx
         ldx bank
         jsr $ff74
STAFAR:  ldx    xx
         stx $02b9
         ldx bank
         jsr $ff77
CMPFAR:  ldx    xx
         stx £02c8
         ldx bank
         jsr $ff7a
```

JSRFAR & JMPFAR

These routines provide an extension to the standard JMP and JSR commands found in 8502. The kernal routines don't have any extra code before they call the routines stored in $0000 – $0400, unlike the STAFAR and LDAFAR routines, so the use is basically the same. These routines make use of addresses $02 – $09 in zero page to pass all the possible required information internal to the CPU, such as the registers. The memory has the following layout:-

$02 – Bank number.
$03 – High byte of address to either JMP or JSR.
$04 – Low byte part.
$05 – Processor status.
$06 – Accumulator.
$07 – X index register.
$08 – Y index register.
$09 – Stack Pointer.

We see that the address contained in locations $03 & $04 are in a different order to what would normally be expected, so for example to jump to addres $456F then set:-

$03 = $45
$04 = $6F

Apart from location $09, the JSRFAR and JMPFAR require all the above information to be setup.
Set up as above then:-
JMPFAR: jsr $ff71 or $02e3
JSRFAR: jsr $ff6e or $02cd
(Both calls do exactly the same actions.)

On the return after a JSRFAR routine, the values in addresses $05-$09 will contain relevant information about the state of the internal registers upon exit of the subroutine called. This information follows the layout described above.

May occur when you want to add a modification to Basic, and you want to sit the program above Basic variable storage which is in RAM 1. If at any time the program needs to call a routine say in the kernal, it can't!

To try the solve the problem, we need to examine the code more closely: The subroutine call to JMPFAR means that when a return is encountered in the code called, it returns back

JSRFAR code:

```
$02cd 20 e3 02    jsr $02e3     call JMPFAR routine.
$02d0 85 06       sta $06       Save returned A,X,Y.
$02d2 86 07       stx $07
$02d4 84 08       sty $08
$02d6 08          php           Save Processor status.
$02d7 68          pla
$02d8 85 05       sta $05
$02da ba          tsx
$02db 86 09       stx $09       Save the stack pointer.
$02dd a9 00       lda   $00     set configuration
$02df 8d 00 ff    sta $ff00     In this case BANK 15.
$02e2 60          rts           return to caller
```

JMPFAR

```
$02e3 a2 00       ldx   $00     Place the address and
$02e5 b5 03       lda $03,x     Processor status on the stack.
$0237 48          pha
$02e8 e8          inx
$02e9 e0 03       cpx   $03
$02eb 90 f8       bcc $02e5
$02ed a6 02       ldx $02       Get Bank.
$02ef 20 6b ff    jsr $ff6b     work out configuration.
$02f2 8d 00 ff    sta $ff00
$02f5 a5 06       lda $06       Get the values of A,X,Y.
$02f7 a6 07       ldx $07
$02f9 a4 08       ldy $08
$02fb 60          rti           return to address on stack.
```

These routines would have been very useful if Commodore hadn't made one mistake. Even if you call these routines in the common area and not via the KERNAL jump, the code then tries to call a kernal routine to convert the bank number to the configuration value. This means to use these routines you must have a configuration where the kernal is visible to the CPU. It also restricts you to only using 16 possible configurations.

An example of this problem can be seem when writing a machine code program which will sit in RAM 1 at high storage say $F000. This situation may occur when you want to add a modification to Basic, and you want to sit the program above Basic variable storage which is in RAM 1. If at any time the program needs to call a routine say in the kernal, it can't!

to the address $23d0, which is common memory.
By looking at the code for these routines given above, we see that the bank conversion call done by both JSRFAR and JMPFAR is actually done by the same piece of code (there's no point looking at the KERNAL versions, as all they do is directly jump to the above routines). Also we see that the JSRFAR routine assumes you want to return to configuration 00, but this isn't always the case. It's obvious that to make these routines more flexible, they need to be modified. The problem is that if they are modified,

will they remain compatable with other codes that also call these routines?

Solution 1

This is just a direct patch on the existing code:

$02ef	from 20 to 2c
$02f2	from 8d to 8e

So the new JMPFAR code is:

```
$02e3 a2 00     ldx  $00      Place the address and
$02e5 b5 03     lda $03,x     Processor status on the stack.
$02e7 48        pha
$02e8 e8        inx
$02e9 e0 03     cpx  $03
$02eb 90 f8     bcc $02e5
$02ed a6 02     ldx $02       Get Bank.
$02ef 20 6b ff  bit $ff6b     Perform no relevant action.
$02f2 8d 00 ff  stx $ff00     Save configuration.
$02f5 a5 06     lda $06       Get the values of A,X,Y.
$02f7 a6 07     ldx $07
$02f9 a4 08     ldy $08
$02fb 60        rti           return to address on stack with the
                              processor status.
```

You will have to put the contents of location $ff00 into $02de before you call the routine. This provides the configuration you want the routine to return to after the JSRFAR.

If you change the code in this manner then you must specify the configuration in location $02 before calling. Because of this it is no longer compatable with the original version, and any other call to this routine expecting the usual code is liable to crash. The interrupt routines don't use this part of the code so there isn't any problem. If, however, you are calling Basic or KERNAL routines they may use these routines and it won't work in this modified form. An alternative solution could be:

Solution 2

$03E4 PHP	
$03E5 PHA	Save registers that could be affected.
$03E6 LDA $FF00	Store current configuration.
$03E9 STA $FE	
$03EB LDA ??	Store New configuration.
$03ED STA $FF00	Patched by calling routine.
$03F0 PLA	Retrieve registers.
$03F1 PLP	
$03F2 JSR $????	Call routine (address to be patched).
$03F5 PHP	Return of routine called.
$03F6 PHA	Save possible affected registers.
$03F7 LDA $FE	
$03F9 STA $FF00	get back old configuration.
$03FC PLA	Retrieve registers.
$03FD PLP	
$03FE RTS	return

This routine mimicks a JSRFAR call. To use it, the following information must be set up.

$03EC	configuration to JSR to
$03F3	low address of routine to call
$03F4	high addrress of routine to call

For a JMPFAR routine, the code becomes:

$03E4 PHP	
$03E5 PHA	
$03E6 LDA ??	– configuration to be patched by caller
$03E8 PLA	
$03E9 PLP	
$03EA JMP $????	– address to be patched by caller

The routine here requires that:
$03EF – configuration
$03EB – low address to jump to
$03EC – high address to jump to

Calling this routine won't affect any of the registers, and the same rule applies on return in the case of the JSRFAR routine. The disadvantage of this system is that code is overwritten from address $03F0-$03FC when the JSRFAR routine is stored. The code removed is a DMA routine which is used for initialising external memory access. If ever a need for the JSRFAR code was required, you could patch it in, execute it, then patch back the DMACALL routine either by saving it first using a program, or by just knowing what it is and putting it back. This code isn't used in the normal running of the system. You should obviously select a method that's suited to your needs.

One final thing to note. These routines stored in the area $0000 – $03ff are crucial for the programmer to be able to access other blocks of storage.

Practical uses

We've now covered what registers are available and what software is available for use. But what can you do with it?

With the upgrade to the system software, you can now design programs to run in any RAM block accessing any part of the available commodore memory. Before, you may have thought you were restricted to just placing programs at certain points because you required access to certain ROMS. I've designed programs which require access to RAM 0, but couldn't actually be in RAM 0 in case of corruption of the data stored there. An example of this is the LABEL-LINKER featured recently in *Your Commodore*. This used the available routines of LDAFAR and STAFAR, along with the modified JSRFAR and JMPFAR of solution 1.

Other possibilities

With so much control over the memory, all sorts of new techniques can be used, and so new programs can be created. The ability to move the location of page zero and page 1 should also broaden your computing horizons considerably. You could move the page 1 and store information in memory via the PHA much faster than the STA command ever could.

One possibility is the concept of multitasking machine code programs making use of the ability to move page 0 and page 1.

128 Corner

Our regular news letter and general comment spot for C128 owners

Welcome to the latest instalment of C128 corner. This is the page where C128 owners can find news of new products, general product news and of course letters, comments and general queries from other readers. Don't forget, if YOU have a query, a bit of news or would just like to make a general comment about the C128 then please do write in. If you don't, then C128 Corner will be incredibly hard to produce, so help us to help you by writing in.
First a letter:

Resetting the C128

In a recent review of the Super Snapshot cartridge (*Your Commodore,* April 1989), the author noted that there was no switch to turn it off. So to switch his C128 from 64 to 128 mode, he had to switch off his machine and remove the cartridge. This is also the case with the Action Replay cartridge and I imagine many others, much to the annoyance of C128 owners everywhere.

While it is technically possible to add a switch that would switch out a cartridge on the C128, without extra circuitry to protect the computer flicking the switch with the power on could damage the computer. Perhaps anything that adds a few pounds to the cost of the cartridge purely for the benefit of C128 owners, isn't considered viable by the manufacturers!

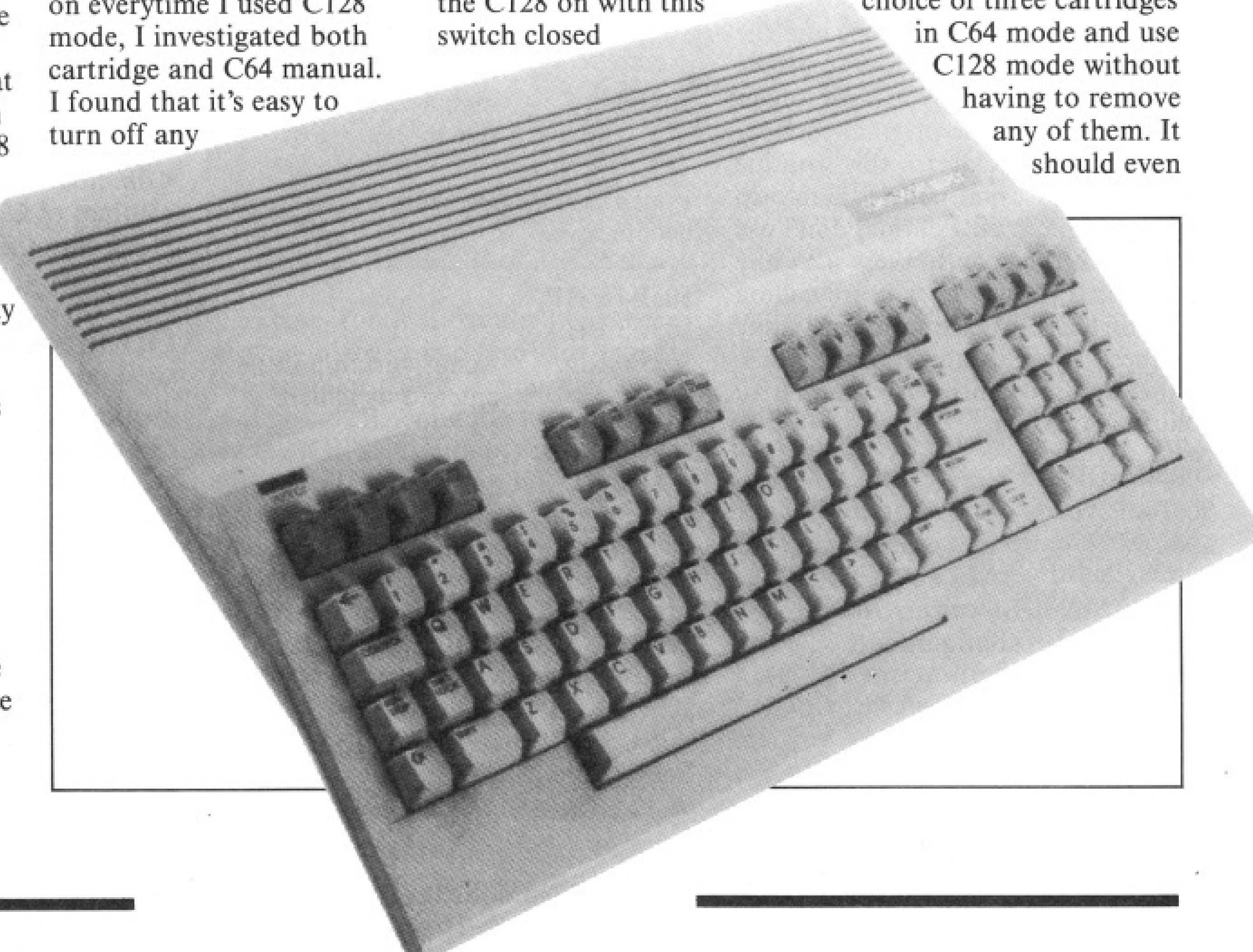
I saw the Datel advert for their motherboard – 'switch out any slot', it says. Great, I thought, just what I need for my C128/Action replay. But, after buying this, I discovered that while it does switch from one cartridge to another, it does it by switching the 5V power supply to the cartridge on and off. It doesn't allow you to go from C64 mode to C128 mode, even with the board switched to an empty slot.

After a period of switching off, inplugging the cartridge and switching on everytime I used C128 mode, I investigated both cartridge and C64 manual. I found that it's easy to turn off any cartridge by simply adding a second switch to the motherboard.

Pin 9, marked EXROM in the C64 manual, is the one that locks the 128 into 64 mode. Inserting a switch into this track on the motherboard would make it possible to switch out the motherboard and cartridges. On Datel's board, the track from pin 9 conveniently goes to the underside of the board for a short distance. At this point I scraped away part of the track, soldered two small wires to each end of the cut track, and connected these to a small switch mounted beside the board's reset button – making pin 9 switchable. Now switching the C128 on with this switch closed allows the use of cartridges in C64 mode as normal. But turning the computer on with the switch open means it powers up in C128 mode without having to remove any cartridges!

Datel's board has some protection fitted to protect the computer/cartridge when switching between slots, and this may mean that turning the computer off when switching between modes isn't strictly necessary – I haven't had any problems anyway.

Switching in this way may not be in any of the manuals, but it is cheap, requires only minimal soldering, and has been used by me for a few months, letting me have the choice of three cartridges in C64 mode and use C128 mode without having to remove any of them. It should even



be possible to add a switch to a cartridge connected across a cut track from PIN 9, though in this case you should turn off the computer before moving the switch.

Pin 9 is located on the user port, and is the 9th from the left on the top row looking from the front of the computer – check your C64 manual for more information.

Mike Vine, Dagenham.

It does seem a great idea to modify your motherboard and cartridges in this way. If you were really adventurous, you could break the track to pin 9 inside your C128 and put the switch inside the computer. However, be warned – we have been unable to try any of the above alterations out, and don't forget modifying any item as indicated would invalidate any warranty. Make the modifications at your own risk.

If anyone else has similar modifications that can be made to C128s, please let us know so that we can pass the information on.

On the subject of cartridges, you may like to hear about the only cartridge we know of that's available for use in C128 mode. The cartridge is the Warp Speed cartridge, available from those Gurus of everything to do with the C128 – Financial Systems Software. The cartridge offers turbo disk loading, a machine code monitor and a host of other features. Contact FSSL on (0863) 553153 for more details.

C128 In The PD

C128 owners often moan about the lack of software available for their computer. Most of the software houses probably haven't even heard of this wonderful machine. If you think that there isn't much 128 software, think again – there's loads of it in the public domain.

Public Domain software is a collection of programs written by various authors with no desire to make money from their labours. The software can be copied and given to other C128 owners, as long as any information that the program's author wishes to be passed on with the program is.

We have recently been contacted by Kingsway Computer Services with details of their 128 public domain library. Prices are extremely cheap, and there are plenty of disks crammed full of available programs.

For a free catalogue contact:

Kingsway Computer Services
140 Rushdale Road
Sheffield
S8 9QE
Tel: (0742) 588429

128 into Amiga will go

Late news just in indicates that a program is now available from FSSL that allows files to be transferred to and from the Commodore Amiga, Atari ST and Apple Macintosh to a C128D or C128 with 1571 disk drive. The package costs £59.95. Please note that this is a file transfer utility only – it does not mean that programs from the aforementioned machines can be used on your C128 or vice versa. The main use for such a utility would be for transferring wordprocessor text files.

Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from *Your Commodore*. Your fingers reach for the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little but slips through unnoticed.

The *Your Commodore* Software Service makes available all of the programs from each issue on both cassette and disk at a price of £6.00 for disk and £4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFONET LTD, 5 River Park Estate, Berkahmsted, Herts, HP4 1HL.
Tel: (04427) 76661

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied as totally working versions, i.e. when possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strap containing the article type, C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strap. The symbols have the following meaning:

This symbol means that the program is available on cassette.

These programs are available on disk.

Please Note

Since the programs supplied on cassette are total working versions of the program, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

JANUARY 1989

PREFAB SPRITES – A powerful sprite editor for the C64.
DAZPRO – A simple but helpful text processor for the C64. Available on disk and cassette but will only store files on tape.
UDG COMPRESSOR – Save on memory when using UDG's in your programs. For C64 only.
WILLIAM TELL – Our popular arcade game for the C64.
+4 AUTORUN – Improve tape loading on your Plus/4 cassette. Only available on disk.
MINIBASE – A database for C128 owners.

ORDER CODE
DISK YDJAN89 £6.00
TAPE YCJAN89 £4.00

FEBRUARY 1989

TAPE MENU – Add a menu system to your program cassettes (C64).
SONIC EFFECTS – A superb sound editor for the C64.
F DUMP – Dump your C64 text screens to printer with ease.
DATA LOADER – A simple way to enter those reams of C64 DATA lines.
SPRITE LIBRARY – A collection of birds to your growing library (C64).
PLAY THE GAME – A superb fruit machine programe for the Plus/4. (Available on disk only).

ORDER CODE
DISK YDFEB89 £6.00
TAPE YCFEB89 £4.00

MARCH 1989

PERSONAL FILE – A cross between a wordprocessor and a database that allows you to set up "cards" that can be quickly altered (C64 Disk only).
LETTER WRITER – An 80 column text editor for writing those personal letters (C64).
BASIC WORKSHOP – A single key entry system, just like a Speccy! (C64)
HEAD FOR HOME – Our version of a popular board game for C16 and Plus/4 owners – available on disk only.
SPRITE LIBRARY – Geometric shapes form this month's installment (C64).
ELECTRONIC NOTEBOOK – A personal diary on disk (C64 disk only).
WILLIAM TELL – Our very popular arcade adventure for the C64.

ORDER CODE
DISK YDMAR89 £6.00
TAPE YCMAR89 £4.00

APRIL 1989

BASEX – Give your C64 new sound, graphics and toolkit commands as well as a machine code assembler with this Basic extension.
AUTOSCROLL – Professional text screen scrolling with this C64 utility.
BALANCE SHEET – Keep your bank manager happy by keeping better track of your money with this C64 program (disk only).
ICON 64 – Add icon control to your own programs with this C64 utility.
80 COLUMNS – A suite of Hi-res graphic commands for C128 80 column users.
ZEUS – A disk file copier for Plus/4 users. Available on disk only.
MULTI-PRECISION ARITHMETIC – A suite of powerful mathematical routines expanding your C64's number crunching abilities.
STORAGE SPACE – Use the RAM behind the C64's Kernal and Basic ROMs for data storage (available on disk only).

ORDER CODE
DISK YDAPR89 £6.00
TAPE YCAPR89 £4.00

MAY 1989

ANTI-FREEZE – Protect your C64 programs from cartridge-based freezers.
MULTI-COLOURED LIST – Brighten up your C64 Basic listings by adding on-screen colour.
PRINTFX – A suite of powerful screen printing commands to add to your Plus/4's Basic. (disk only).
BANKER 128 – Money Management for C128 disk users.
TURBO TOUCH – A superb typing trainer for C64 disk users.

ORDER CODE
DISK YDMAY89 £6.00
TAPE YCMAY89 £4.00

Cassettes or disks are available from March 1986. Please ring the editorial office (01-437 0626) for details of these.

ORDER FORM – PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
AUGUST '89		TAPE (£4.00)	YC AUG'89	
AUGUST '89		DISK (£6.00)	YD AUG'89	
OVERSEAS POST £1				
			TOTAL	

NAME..

ADDRESS..

POSTCODE..

I enclose a cheque/postal order for £....... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.
All orders should be sent to: *YOUR COMMODORE,*
READERS SERVICES, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS. HP2 7ST

Please allow 28 days for delivery.

Mouse 80

In the July issue of **Your Commodore*****, we featured D.H. Faber's highly useful 80-column mode mouse utility for the 128. Well, here's the listings we didn't have room for last month***

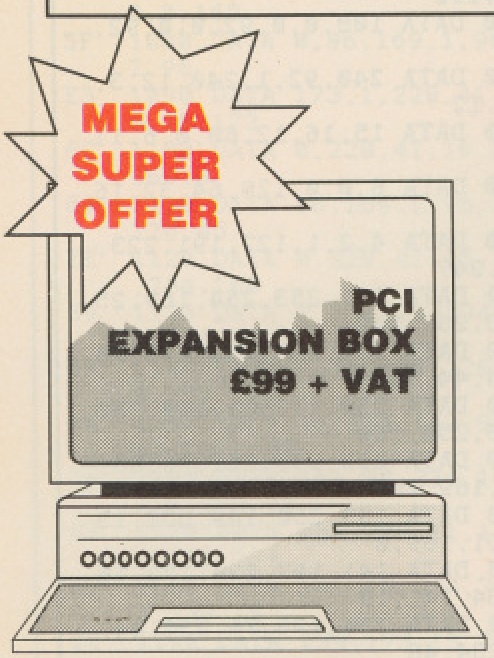
```
    1,16  40
F2  1060 DATA 17,173,0,220,41,1,
    240,13
57  1070 DATA 173,0,220,41,16,24
    0,3,169
5F  1080 DATA 0,96,169,1,96,169,
    2,96
E3  1090 DATA 173,1,220,41,16,24
    0,10,173
6C  1100 DATA 0,220,41,16,240,3,
    169,0
89  1110 DATA 96,169,1,96,173,1,
    220,45
EE  1120 DATA 0,220,41,15,141,74
    ,234,169
69  1130 DATA 0,141,19,234,173,1
    4,234,208
55  1140 DATA 3,76,228,226,32,13
    8,230,141
4A  1150 DATA 92,234,142,93,234,
    32,171,230
24  1160 DATA 141,94,234,142,95,
    234,173,19
89  1170 DATA 234,208,11,169,1,1
    41,16,234
BD  1180 DATA 141,15,234,76,8,22
    7,238,15
26  1190 DATA 234,173,15,234,201
    ,9,48,20
3A  1200 DATA 169,1,141,15,234,2
    38,16,234
01  1210 DATA 173,16,234,201,5,4
    8,5,169
4C  1220 DATA 4,141,16,234,173,9
    2,234,141
0E  1230 DATA 96,234,173,93,234,
    141,97,234
89  1240 DATA 173,94,234,141,98,
    234,173,95
B9  1250 DATA 234,141,99,234,174
    ,16,234,202
7D  1260 DATA 240,77,24,173,96,2
    34,109,92
DE  1270 DATA 234,141,92,234,173
    ,97,234,109
7A  1280 DATA 93,234,141,93,234,
    24,173,98
20  1290 DATA 234,109,94,234,141
    ,94,234,173
AD  1300 DATA 99,234,109,95,234,
    141,95,234
FD  1310 DATA 76,184,226,173,25,
    212,172,20
7C  1320 DATA 234,32,241,227,140
    ,20,234,141
04  1330 DATA 92,234,142,93,234,
    173,26,212
01  1340 DATA 172,21,234,32,241,
    227,140,21
5B  1350 DATA 234,141,94,234,142
    ,95,234,24
D6  1360 DATA 173,92,234,109,104
    ,234,141,104
F6  1370 DATA 234,173,93,234,109
    ,105,234,141
82  1380 DATA 105,234,173,105,23
    4,16,11,169
52  1390 DATA 0,141,104,234,141,
    105,234,76
FA  1400 DATA 69,227,173,105,234
    ,201,1,48
97  1410 DATA 19,208,7,173,104,2
    34,201,64
05  1420 DATA 144,10,169,63,141,
    104,234,169
4B  1430 DATA 1,141,105,234,56,1
    73,109,234
A8  1440 DATA 237,94,234,141,109
    ,234,173,110
62  1450 DATA 234,237,95,234,141
    ,110,234,173
40  1460 DATA 110,234,16,11,169,
    0,141,109
2E  1470 DATA 234,141,110,234,76
    ,174,227,169
43  1480 DATA 0,141,93,234,173,2
    2,234,141
7C  1490 DATA 92,234,14,92,234,4
    6,93,234
9E  1500 DATA 14,92,234,46,93,23
    4,14,92
B3  1510 DATA 234,46,93,234,173,
    92,234,208
DD  1520 DATA 3,206,93,234,206,9
    2,234,173
0B  1530 DATA 110,234,205,93,234
    ,48,22,208
ED  1540 DATA 8,173,109,234,205,
    92,234,144
C1  1550 DATA 12,173,92,234,141,
    109,234,173
2C  1560 DATA 93,234,141,110,234
    ,173,104,234
E5  1570 DATA 141,92,234,173,105
    ,234,141,93
FD  1580 DATA 234,78,93,234,110,
    92,234,78
C1  1590 DATA 93,234,110,92,234,
    173,92,234
6E  1600 DATA 141,9,234,173,109,
    234,141,92
F8  1610 DATA 234,173,110,234,14
    1,93,234,78
DA  1620 DATA 93,234,110,92,234,
    78,93,234
9E  1630 DATA 110,92,234,78,93,2
    34,110,92
24  1640 DATA 234,173,92,234,141
    ,10,234,96
BC  1650 DATA 140,91,234,141,90,
    234,162,0
6B  1660 DATA 56,237,91,234,41,1
    27,201,64
31  1670 DATA 176,10,74,240,24,1
    72,90,234
65  1680 DATA 141,19,234,96,9,19
    2,201,255
E7  1690 DATA 240,11,56,106,162,
    255,172,90
01  1700 DATA 234,141,19,234,96,
    169,0,96
CD  1710 DATA 173,76,234,44,76,2
    34,16,19
4F  1720 DATA 173,10,234,24,105,
    2,205,22
9A  1730 DATA 234,48,23,169,127,
    45,76,234
A3  1740 DATA 141,76,234,173,10,
    234,201,2
5D  1750 DATA 16,8,169,128,13,76
    ,234,141
34  1760 DATA 76,234,173,76,234,
    44,76,234
A5  1770 DATA 112,18,173,9,234,2
    01,78,48
8A  1780 DATA 26,169,64,13,76,23
    4,141,76
B3  1790 DATA 234,76,116,228,173
    ,9,234,201
B0  1800 DATA 3,16,8,169,191,45,
    76,234
6A  1810 DATA 141,76,234,174,9,2
    34,173,76
61  1820 DATA 234,44,76,234,80,2
    ,202,202
0B  1830 DATA 142,101,234,174,10
    ,234,44,76
B1  1840 DATA 234,48,1,202,142,1
    06,234,169
0F  1850 DATA 0,141,93,234,173,1
    01,234,141
B8  1860 DATA 92,234,14,92,234,4
    6,93,234
0C  1870 DATA 14,92,234,46,93,23
    4,173,104
69  1880 DATA 234,141,94,234,173
    ,105,234,141
08  1890 DATA 95,234,56,173,94,2
    34,237,92
35  1900 DATA 234,141,94,234,173
    ,95,234,237
67  1910 DATA 93,234,141,95,234,
    173,94,234
11  1920 DATA 10,141,103,234,173
    ,76,234,44
56  1930 DATA 76,234,80,9,173,10
    3,234,56
E0  1940 DATA 233,12,141,103,234
    ,169,0,141
42  1950 DATA 93,234,173,106,234
    ,141,92,234
3C  1960 DATA 14,92,234,46,93,23
    4,14,92
8D  1970 DATA 234,46,93,234,14,9
    2,234,46
A7  1980 DATA 93,234,173,109,234
    ,141,94,234
3B  1990 DATA 173,110,234,141,95
    ,234,56,173
54  2000 DATA 94,234,237,92,234,
    141,94,234
33  2010 DATA 173,95,234,237,93,
    234,141,95
73  2020 DATA 234,173,94,234,141
    ,108,234,173
4B  2030 DATA 76,234,44,76,234,1
    6,1,96
6C  2040 DATA 56,173,108,234,233
    ,7,141,108
44  2050 DATA 234,96,169,0,141,1
    07,234,24
BC  2060 DATA 173,106,234,109,10
    7,234,141,18
07  2070 DATA 234,169,0,141,102,
    234,24,173
A5  2080 DATA 101,234,109,102,23
    4,141,17,234
87  2090 DATA 174,73,234,202,48,
    68,138,10
50  2100 DATA 168,173,17,234,217
    ,224,233,208
77  2110 DATA 242,173,18,234,217
    ,225,233,208
79  2120 DATA 234,206,73,234,189
    ,77,234,72
5B  2130 DATA 189,248,233,72,232
    ,236,73,234
A5  2140 DATA 48,2,208,101,138,1
    0,168,189
4F  2150 DATA 77,234,157,76,234,
    189,248,233
0C  2160 DATA 157,247,233,185,22
    4,233,153,222
C7  2170 DATA 233,185,225,233,15
    3,223,233,76
A6  2180 DATA 117,229,174,17,234
    ,172,18,234
08  2190 DATA 32,204,230,165,250
    ,133,252,165
A5  2200 DATA 251,133,253,24,173
    ,23,234,101
72  2210 DATA 252,133,252,173,24
    ,234,101,253
D3  2220 DATA 133,253,165,252,16
    6,253,32,37
15  2230 DATA 224,72,165,250,133
    ,252,165,251
FB  2240 DATA 133,253,24,173,4,2
    34,101,252
```

```
19 2250 DATA 133,252,173,5,234,101,253,133
96 2260 DATA 253,165,252,166,253,32,37,224
D5 2270 DATA 72,174,73,234,104,157,248,233
E9 2280 DATA 104,157,77,234,138,10,168,173
F8 2290 DATA 17,234,153,224,233,173,18,234
1E 2300 DATA 153,225,233,238,73,234,173,6
A5 2310 DATA 234,133,250,173,7,234,133,251
D4 2320 DATA 189,248,233,16,15,24,173,220
4F 2330 DATA 233,101,250,133,250,173,221,233
60 2340 DATA 101,251,133,251,169,0,133,253
27 2350 DATA 189,77,234,133,252,160,3,6
4C 2360 DATA 252,38,253,136,16,249,24,165
A0 2370 DATA 252,101,250,133,250,165,253,101
F8 2380 DATA 251,133,251,169,25,133,252,169
4D 2390 DATA 234,133,253,173,107,234,10,109
1E 2400 DATA 107,234,109,102,234,10,10,10
AD 2410 DATA 24,101,252,133,252,144,2,230
D6 2420 DATA 253,165,250,166,251,32,45,224
80 2430 DATA 160,0,169,31,32,15,224,145
8B 2440 DATA 252,200,192,8,48,244,238,102
3A 2450 DATA 234,173,102,234,201,3,16,3
C3 2460 DATA 76,71,229,238,107,234,173,107
49 2470 DATA 234,201,2,16,3,76,56,229
6C 2480 DATA 96,162,0,160,0,173,74,234
43 2490 DATA 41,4,240,15,173,74,234,41
58 2500 DATA 8,240,2,152,96,169,1,141
B0 2510 DATA 19,234,96,169,255,170,141,19
20 2520 DATA 234,96,162,0,160,0,173,74
A1 2530 DATA 234,41,1,240,16,173,74,234
78 2540 DATA 41,2,240,2,152,96,169,255
31 2550 DATA 170,141,19,234,96,169,1,141
AA 2560 DATA 19,234,96,169,0,133,251,152
FF 2570 DATA 10,10,133,250,6,250,38,251
FB 2580 DATA 6,250,38,251,165,250,141,94
BE 2590 DATA 234,165,251,141,95,234,6,250
B3 2600 DATA 38,251,6,250,38,251,24,173
23 2610 DATA 94,234,101,250,133,250,173,95
A1 2620 DATA 234,101,251,133,251,138,24,101
05 2630 DATA 250,133,250,144,2,230,251,96
3C 2640 DATA 173,73,234,201,7,48,119,174
62 2650 DATA 224,233,172,225,233,32,204,230
4F 2660 DATA 165,250,133,252,165,251,133,253
8A 2670 DATA 24,173,4,234,101,252,133,252
D7 2680 DATA 173,5,234,101,253,133,253,172
9F 2690 DATA 248,233,165,252,166,253,32,27
75 2700 DATA 224,165,250,133,252,165,251,133
68 2710 DATA 253,24,173,23,234,101,252,133
B7 2720 DATA 252,173,24,234,101,253,133,253
37 2730 DATA 172,77,234,165,252,166,253,32
32 2740 DATA 27,224,162,0,232,236,73,234
46 2750 DATA 240,30,189,77,234,157,76,234
9A 2760 DATA 189,248,233,157,247,233,138,10
A1 2770 DATA 168,185,224,233,153,222,233,185
49 2780 DATA 225,233,153,223,233,76,93,231
4F 2790 DATA 206,73,234,76,9,231,169,25
26 2800 DATA 133,250,169,234,133,251,173,222
74 2810 DATA 233,133,252,173,223,233,133,253
DD 2820 DATA 173,222,233,141,92,234,173,223
EB 2830 DATA 233,141,93,234,6,252,38,253
0C 2840 DATA 6,252,38,253,6,252,38,253
9C 2850 DATA 6,252,38,253,24,173,6,234
38 2860 DATA 101,252,133,252,173,7,234,101
F9 2870 DATA 253,133,253,169,5,141,17,234
39 2880 DATA 160,0,165,252,166,253,32,45
60 2890 DATA 224,177,250,170,169,31,32,3
03 2900 DATA 224,200,192,8,48,243,169,8
F3 2910 DATA 24,101,250,133,250,144,2,230
8A 2920 DATA 251,169,16,24,101,252,133,252
2E 2930 DATA 144,2,230,253,206,17,234,16
63 2940 DATA 207,169,0,141,18,234,173,18
D5 2950 DATA 234,10,109,18,234,141,107,234
3C 2960 DATA 109,107,234,168,190,224,233,185
ED 2970 DATA 225,233,168,32,204,230,165,250
EE 2980 DATA 133,252,165,251,133,253,24,173
4A 2990 DATA 4,234,101,250,133,250,173,5
FE 3000 DATA 234,101,251,133,251,165,250,166
6F 3010 DATA 251,32,45,224,160,0,152,24
0C 3020 DATA 109,107,234,170,173,100,234,208
42 3030 DATA 23,173,223,233,240,9,189,248
D7 3040 DATA 233,9,128,72,76,93,232,189
72 3050 DATA 248,233,41,127,72,76,93,232
3E 3060 DATA 189,248,233,72,104,170,169,31
B1 3070 DATA 32,3,224,200,192,3,48,206
0B 3080 DATA 24,173,23,234,101,252,133,252
02 3090 DATA 173,24,234,101,253,133,253,165
2C 3100 DATA 252,166,253,32,45,224,160,0
EA 3110 DATA 152,24,109,107,234,170,173,100
DE 3120 DATA 234,208,10,173,92,234,72,238
F4 3130 DATA 92,234,76,154,232,189,77,234
61 3140 DATA 72,104,170,169,31,32,3,224
75 3150 DATA 200,192,3,48,219,238,18,234
81 3160 DATA 173,18,234,201,2,48,1,96
39 3170 DATA 76,255,231,7,0,0,7,126
70 3180 DATA 0,7,124,0,12,126,0,12
4F 3190 DATA 71,128,12,1,224,108,0,112
5F 3200 DATA 108,0,0,92,0,0,92,3
98 3210 DATA 240,92,1,240,12,3,240,12
2F 3220 DATA 15,16,12,60,0,6,112,0
B0 3230 DATA 6,0,0,128,64,32,16,8
1D 3240 DATA 4,2,1,127,191,223,239,247
1F 3250 DATA 251,253,254,169,25,133,252,169
F5 3260 DATA 234,133,253,173,76,234,44,76
57 3270 DATA 234,112,11,169,180,133,250,169
61 3280 DATA 232,133,251,76,23,233,169,204
B9 3290 DATA 133,250,169,232,133,251,169,0
35 3300 DATA 141,107,234,173,107,234,141,18
19 3310 DATA 234,173,76,234,44,76,234,48
CB 3320 DATA 9,56,169,7,237,107,234,141
0D 3330 DATA 18,234,24,173,107,234,109,108
6E 3340 DATA 234,201,8,48,3,24,105,16
DA 3350 DATA 141,92,234,173,18,234,10,109
56 3360 DATA 18,234,141,93,234,168,177,250
00 3370 DATA 41,240,74,74,74,74,141,89
4F 3380 DATA 234,177,250,41,15,141,75,234
95 3390 DATA 172,93,234,200,173,89,234,201
05 3400 DATA 8,48,4,56,233,8,200,170
5A 3410 DATA 177,250,61,228,232,72,173,13
BD 3420 DATA 234,240,4,104,169,0,72,24
46 3430 DATA 173,89,234,109,103,234,141,17
E7 3440 DATA 234,173,92,234,141,94,234,173
EA 3450 DATA 17,234,201,8,48,18,56,233
2C 3460 DATA 8,141,17,234,24,173,94,234
3A 3470 DATA 105,8,141,94,234,76,144,233
EA 3480 DATA 172,94,234,174,17,234,104,208
77 3490 DATA 8,177,252,61,236,232,76,191
1B 3500 DATA 233,177,252,29,228,232,145,252
F1 3510 DATA 238,89,234,173,89,234,205,75
67 3520 DATA 234,48,149,240,147,238,107,234
D5 3530 DATA 173,107,234,201,8,48,1,96
A8 3540 DATA 76,28,233,0,16,244,0
```

MEGALAND 0703 332225

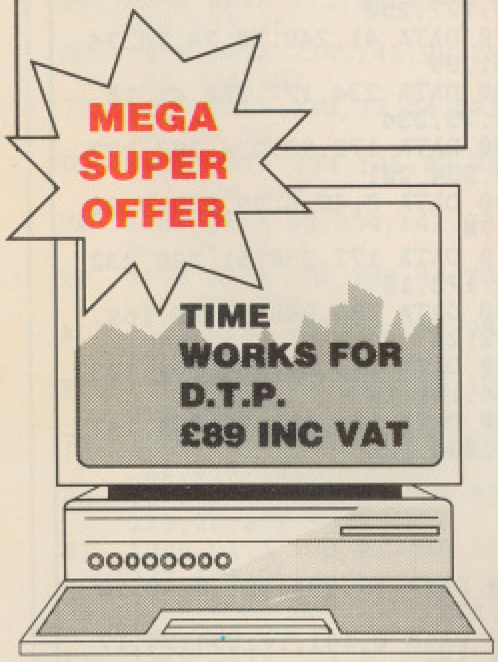
Specialists in Mail Order

WHY MEGALAND • 24 Hour Delivery • 12 months warranty on all hardware • On site maintenance optional • Technical support • 24 Hour ordering service • Special price offers • MEGASOFT Club membership for extra special OFFERS



SPECIAL OFFER

PCI COMPUTER + HIGH RES MONITER ONLY £309 INC VAT



NOW IN STOCK
AMIGA A500 AIR MILE PACK
AMIGA COMPUTER,TV MOD
NEBULAS,SPITZ PAINT
ROGER RABBIT ,STAR RAY
+500 AIR MILES 499 inc.
WHEN YOU BUY A AIR MILES PACK FROM MEGALAND WE WILL GIVE YOU £25 OFF YOUR NEXT PURCHASE OR £25 TOWARDS YOUR HOLIDAY

AMSTRAD PC 2086

PC2086 SD12 MD	£619 inc
PC2086 SD 14 CD	£775 inc
PC 2086 12 HRCD	£875 inc
PC 2086 SD 14 HRCD	£975 inc
PC 2086 DD 12 MD	£775 inc
PC 2086 DD 14 CD	£925 inc
PC 2086 DD 12 HRDC	£1025 inc
PC 2086 DD 14 HRCD	£1125 inc
PC 2086 HD 12 MD	£1025 inc
PC 2086 HD 14 CD	£1175 inc
PC 2086 12 HRCD	£1285 inc
PC 2086 14 HRDC	£1395 inc

AMSTRAD PCW

PCW 8256	£389 inc
PCW 8512	£465 inc
PCW 9512	£515 inc

AMSTRAD PPC PORTABLES

PPC 512S	£415 inc
PPC 640S	£515 inc
PPC 512D	£565 inc
PPC 640D	£720 inc

AMSTRAD PC1512

PC1512 SD M/M	£415 inc
PC1512 DD M/M	£515 inc
PC1512 SD C/M	£515 inc
PC1512 DD C/M	£615 inc

AMSTRAD PC1640

PC1640 SD ECD	£720 inc
PC1640 DD ECD	£825 inc
PC1640 HD20 ECD	£1025 inc
PC1640 SD C/D	£619 inc
PC1640 DD C/D	£720 inc
PC1640 HD20 C/D	£925 inc
PC1640 Sd M/D	£515 inc
PC1640 DD M/D	£619 inc
PC1640 HD20 M/D	£825 inc

COMMODORE PRICE REDUCTIONS

FULLY COMPTABILE SINGLE DRIVE PC COMMODORE PC1 £269 + VAT
(includes mono monitor and able-1 integrated software)
AS ABOVE BUT WITH 14 in C.G.A. COLOUR MONITOR £369 + VAT

COMMODORE DESK TOP PCS

ALL XT'S AT 3 SPEEDS 4,77,8,10 Mz supplied with AT-E 102 keyboards & Dos

PC 10 111 Single drive mono monitor	£485 + VAT
Single drive C.G.A. colour monitor	£599 + VAT
Single drive E.G.A. colour monitor	£799 + VAT
PC10 111 Dual drive mono monitor	
Dual drive C.G.A. monitor	£659 + VAY
Dual drive E.G.A. monitor	£789 + VAT
PC20 111 20 MB Hard disk mono monitor	£689 + VAT
20MB Hard disk C.G.A. monitor	£839 + VAT
20MB Hard disk E.G.A. monitor	£949 + VAT

COMMODORE PC40 AT's

Include V.G.A. Adaptor, 1MB RAM 1.2 MB Drive

PC40 111 Sngle drive V.G.A. mono	£1149 + VAT
PC40 111 Single drive V.G.A. colour monitor	£1369 + VAT
PC40 111 40 MB Hard disk V.G.A. mono	£1419 + VAT
PC40 111 40 MB Hard disk V.G.A. Colour monitor	£1579 + VAT
PC40 111 40 MB Hard disk & 40 MB Tape streamer mono	£1629 + VAT
PC40 111 40MB Hard disk & 40 MB Tape colour monitor	£1839 + VAT

COMMODORE PC 60 s 386 DESK TOP

TOP Standard V.G.A. Adaptor 16MZ

PC60 512K RAM 80386 MONO	£1029 + VAT
PC60 512K RAM 80386 C.G.A. COLOUR	£1159 + VAT
PC60 512K RAM 80386 E.G>A. COLOUR	£1199 + VAT
PC60 2.5 MB RAM 80386 MONO	£1649 + VAT
PC60 2.5 MB RAM 80386 C.G.A. COLOUR	£1779 + VAT
PC60 2.5 MB RAM 80386 E.G.A. COLOUR	£1819 + VAT
PC60-40 40MB Hard disk 2.5 MB RAM 80386 MONO	£1939 + VAT
PC60-40 40 MB Hard disk 2.5 MB RAM C.G.A.	£2105 + VAT
PC60-40 MB Hard disk 2.5 MB RAM E.G.A.	£2149 + VAT
AS ABOVE WITH 40 MB TAPE STREAMER	
Mono Monitor	£2229 + VAT
C.G.A. Monitor	£2395 + VAT
E.G.A. Monitor	£2439 + VAT
PC60-80 80MB Hard disk Mono monitor	£2105 + VAT
PC60-80 80MB Hard disk mono monitor	£2269 + VAT
PC60-80 80MB Hard disk E.G.A.	£2319 + VAT
AS ABOVE WITH 40 MB TAPE STREAMER	
Mono monitor	£2394 + VAT
C.G.A. monitor	£2559 + VAT
E.G.A. monitor	£2599 + VAT

ACCOUNTS S WARE

Sage Bookeeper	£99
Sage Accountant	£149
Sage Acountant Plus	£249
Sage Financial Controller	£399
Sage Payroll 2	£149
Migent Accountability	£79

SPREADSHEETS

Amstrad Supercalc 3.2.1.	£62
Ashton Tate Javelin Plus	£245
Borland Quattro	£199
Grafox Logistix	£79
Lotus 1-2-3 Ver 2.01	£395
Lotus Hal	£120
Lotus Report Writer	£79
Microsoft Excel	£395
Microsoft Multiplan Jnr	£56
Sage PC Planner	£99
Supercalc Ver 5	£396
VP Planner	£66
VP Planner	£110

ATARI

520 STFM 1 Mb Drive Explorer Pack	£258.75 inc.
520 STFM Super Pack (21 games and business software)	£389.00 inc.
520 STFM and Mono Monitor	£372.75 inc.
1040 STFM TV Modulator	£399.00 inc.
1040 STFM Super Pack (21 games and business software)	£419.00 inc.
1040 STFM with Microsoft Write and V.I.P. Professional	£419.00 inc.
1040 STFM Super Pack (21 games and business software), Microsoft Write & V.I.P. Professional	£455.00 inc.
1040 STFM with Atari 124 Mono Monitor	£499.00 inc.
1040 STFM with Colour Monitor SC1224	£649.00 inc.
1040 STFM Super Pack with Mono Monitor 124	£425.00 inc.

NOW IN STOCK THE NEW S20 STFM POWER PACK WITH 20 GAMES £389 INC VAT.

42 - 44 MILLBROOK ROAD EAST, SOUTHAMPTON

OPEN MONDAY TO SATURDAY 9.00am to 5.30pm

PROJECT FIRESTART

There seems to be a whole new type of game springing up, games that tell stories but are definitely not adventures in the traditional sense. One such title is *Project Firestart* from Electronic Arts, written by the same team that gave you *Arcticfox* and *Skyfox* II.

Horror is the theme behind the plot, coupled with a setting in outer space. Prometheus is a research ship sponsored by the Space Science Foundation. The experiments conducted on this ship involved genetic manipulation with the aim of producing a form of cheap labourer for various mining organisations. The problem is that the S.S.F. has lost contact with the Prometheus. Something seems to have gone astray and, as their chief troubleshooter, it's up to you to see if you can find out what's gone wrong and restore some sort of order to the mission.

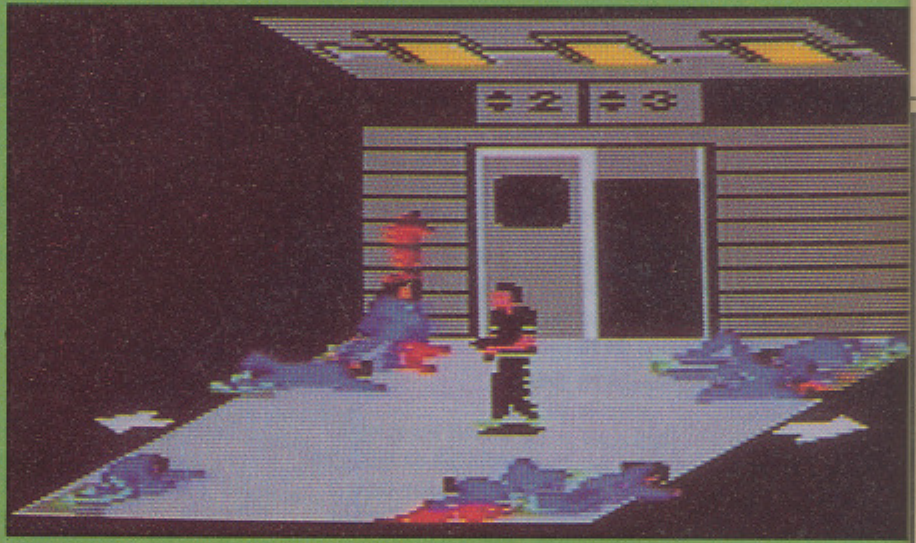
The game is presented as a sort of film. Thus, while you are walking round the ship, exploring, the game will cut to close-ups of various items that it feels are significant. These are usually dead bodies, accompanied by a liberal helping of tomato sauce. Whatever the problem is, it certainly has a violent streak to it.

To keep you in touch with the plot, the scene will also fade to show you what's happening elsewhere on the ship. These scenes are usually connected with a failure in the cryogenic systems – bodies waking up in coffins, and then getting ripped limb from limb a few minutes later.

The ship is presented in a mixture of 2 and 3-D views. As you move close to anything important to the plot, a text window at the bottom of the screen gives you the option to carry out simple commands; pick up an object, turn something on, open a door and so on. This manipulation of objects is crucial to the game, although it will be some time before you have much idea of what's going on.

Hindering your progress are a series of diminutive, but angry, green monsters that appear not to have your best interests at heart. They can be zapped with your laser, but this only has limited power, so you'll need to discover where the weapons are stored. I found this the weakest part of the game, for the simple reason that you often get killed without being able to do anything in self-defence. For example, at one point the story cut to another part of the ship, and when it returned, I was surprised to find several monsters mauling my body! The monsters are best killed at long range – they drain too much of your energy when they touch you, and two ganging up on you with no warning is invariably fatal.

That apart though, *Project Firestart* is an interesting idea, generally well presented. It comes on four sides of disk, so there is a lot of game for your money, even if you do have to wait a long time for new locations to load. The graphics are reasonable, although a bit more description in the text would have added considerably to the atmosphere.



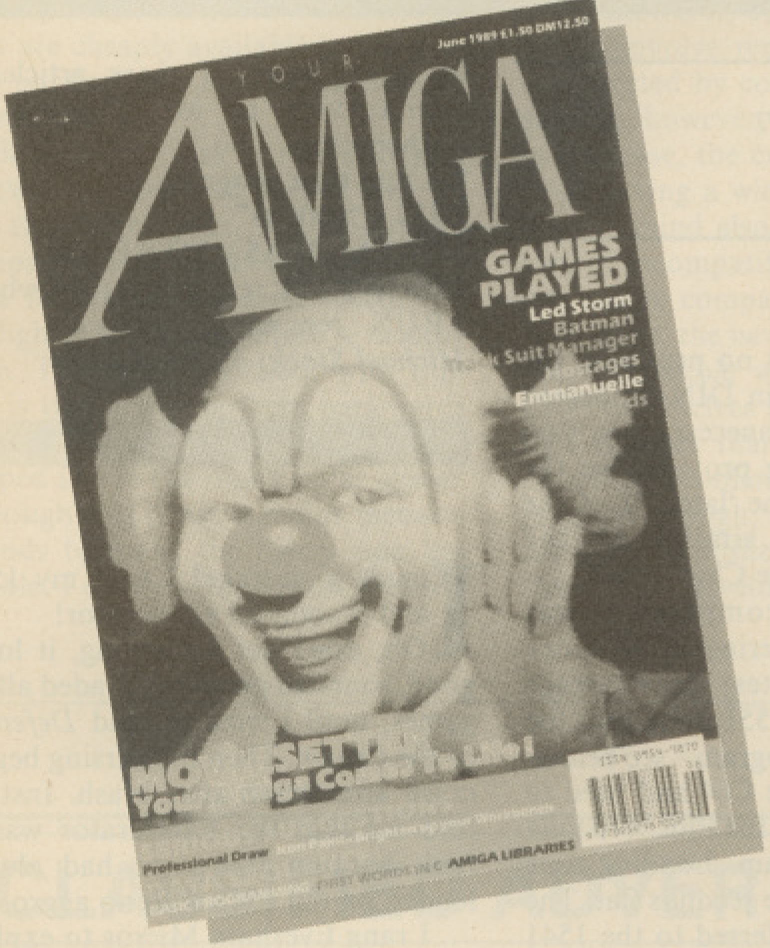
Sound is a mixture of effects and tunes – the program always lets you know when the baddies are approaching! It takes some time to get properly into the game, but it's certainly worthwhile persevering.

Touchline:
Title: *Project Firestart.* ***Supplier:*** *Electronic Arts, 11/49 Station Road, Langley, Berks. SL3 8YN.* ***Price:*** *£14.99 – disk only.*



Have YOUR AMIGA

*delivered to your door FREE !**



That's right, if you take out a year's subscription to Your Amiga we will make sure that it is delivered to your door each month at no extra charge*. Just fill in the coupon below and send it to the address given with a cheque, money order or credit card instructions to cover the cost of the subscription. We'll do the rest.

UK: £23.40; EUROPE: £29.00; MIDDLE EAST: £29.20; FAR EAST: £31.40; REST OF THE WORLD: £29.60 or USA: $52.00

Airmail Rates on Request.

*** Overseas subscription rates include postage.**

Please commence my subscription to Your Amiga with the .... issue. I enclose my cheque/money order for £.......................... made payable to ARGUS SPECIALIST PUBLICATIONS

Or Debit my Access/Visa

No. [ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ][ ]

Valid from .... to ....

Signature ....

Name ....

Address ....

.... Postcode ....

Send this form with your remittance to:
INFONET LTD., 5 River Park Estate, Billet Lane, BERKHAMSTED, Herts. HP4 1HL, United Kingdom

---

NEW FOR YOUR COMMODORE FROM TRILOGIC!

TRILOGIC

AMIGA AUDIO DIGITISER

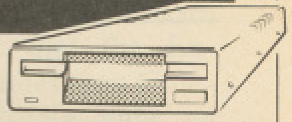


— NEW, CHOOSE MONO OR STEREO VERSION

Both Amiga audio digitisers give superb performance, unsurpassed at the price. No software is supplied, since they are fully compatible with Perfect Sound, Prosound Designer, Audiomaster, and Datel's Prosampler. Sampling rates up to 60KHz are possible depending upon the software. An audio lead is supplied for connecting to the headphone socket or line output of a radio, personal stereo, keyboard etc. Full instructions are included, and the mono version also has an L.E.D. overload indicator. A public domain "Sound Workshop" disk is available which has demo versions of Audiomaster & Perfect Sound etc.

MONO DIGITISER ....**£27.99 POST FREE** STEREO DIGITISER **£37.99**
SOUND WORKSHOP DISK .... **£4.99 POST FREE IF PURCHASED WITH DIGITISER**
ADAPTOR FOR A1000 ....**£2.00**

EXTERNAL 3.5" DISK DRIVES



- 880K FORMATTED CAPACITY
- THROUGH POST
- VERY QUIET OPERATION
- SLIMLINE STEEL CASE
- LOW POWER CONSUMPTION

3.5" DRIVE WITH ON/OFF SWITCH ....**£79.99 POST FREE**

AMIGA MONITOR & PRINTER LEADS

RGB TV & MONITOR LEADS



We have leads to connect all AMIGAS to your TV or colour monitor provided it has an RGB input socket. All leads give a much clearer picture than using the AMIGA MODULATOR, permit ALL 4096 colours to be displayed and include the audio lead (to give stereo with stereo tv's).

ORDER AL 1 FOR TV's WITH 21 PIN EURO (SCART) SOCKET. FITS PHILIPS, FIDELITY, SONY, GRUNDIG, NORMENEDE, ETC. ....**ONLY £9.99**
ORDER AL 2 FOR FERGUSON TV's WITH 7 or 8 PIN DIN SOCKET. MODELS MC01 & MC05, ETC. ....**ONLY £9.99**
ORDER AL 4 FOR HITACHI & GRANADA TV's WITH 7 PIN DIN SOCKET. MODELS CPT1444, ETC. ....**ONLY £9.99**
AMIGA TO MULTISYNC MONITOR (analogue RGB) AL11 ....**£9.99**

MANY OTHER AMIGA MONITOR/TV LEADS LISTED IN OUR FREE CATALOGUE.
OUR LEADS ARE GUARANTEED TO WORK WHERE OTHERS DON'T!
LEADS ALSO AVAILABLE FOR ATARI ST RANGE. PLEASE CONSULT US IF IN DOUBT.

NEW — MONITOR SHARER

SHARE 2 COMPUTERS WITH ONE TV/MONITOR
Plug-in the scart leads from any two computers eg AMIGA & ST, and connect the SHARER to the tv or monitor's scart socket. Just press the switch to instantly select either computer ....**MSWI £15.99**

OTHER LEADS

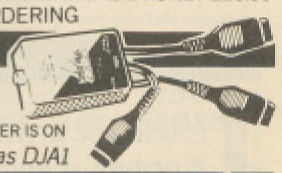
AMIGA 64 EMULATOR LEADS	**AEL 1 ONLY £4.99**
MODULATOR EXTENSION LEADS	**MEL 1 ONLY £8.99**
AMIGA PRINTER LEAD — Parallel Type 1.5M long	**AMP 1 ONLY £6.99**
3.0M long	**AMP 3 ONLY £9.99**
AMIGA A500/BBC EMULATOR LEAD	**BE1 ONLY £7.99**
AMIGA A500 TO SERIAL (RS232) PRINTER 1.5m	**RS1 ONLY £9.99**
AMIGA A500 TO MODEM (25W 'D') 1.5m	**RS6 ONLY £10.99**
AMIGA NULL MODEM LEAD — LINKS TWO A500s 1.5m	**RS5 ONLY £7.99**
AMIGA TO HIFI LEAD 3m 2 PHONO PLUGS EACH END	**HL1 ONLY £4.99**
AMIGA TO HIFI LEAD 5m 2 PHONO PLUGS EACH END	**HL2 ONLY £5.99**

OTHER LENGTHS OF LEAD AVAILABLE — PLEASE SEE OUR CATALOGUE

EXTERNAL DRIVE SWITCH FOR DF1	**ADE3 ONLY £8.99**
EXTERNAL DRIVE SWITCH FOR DF1 & DF2	**ADE4 ONLY £10.99**

PLEASE STATE WHICH AMIGA YOU HAVE WHEN ORDERING

MOUSE/JOYSTICK SWITCHER



- NO MORE UNPLUGGING MOUSE WHEN JOYSTICK REQUIRED
- EXTENDS THE PORT FOR EASIER ACCESS
- SAVES WEAR & TEAR ON THE MOUSE PORT
- PUSHBUTTON SWITCH CAN BE OPERATED WHEN THE COMPUTER IS ON

A VERY USEFUL GADGET — ONLY £10.99 order as DJA1

AMIGA MINI STEREO AMPLIFIERS



MINIAMP 2 combines a mini stereo power amp with two neat speaker units which connect directly to your AMIGA. They are ideal for use with mono TV's & monitors, & simply plug in for instant stereo sound. You'll be amazed at the difference.
MINIAMP 2 WITH REMOTE VOLUME CONTROL & ALL LEADS **ONLY £19.99**
MINIAMP 4 comprises separate 4 watts per channel high quality mini stereo amplifier with loudspeaker/headphone socket, separate speaker units with 4" twin cone drivers, mains power pack & connecting lead for AMIGA. CAN BE USED AS 'WALKMAN' BOOSTER & WITH OTHER COMPUTERS

MINIAMP 4 STEREO AMPLIFIER SPEAKER SYSTEM	**MA4 £34.99**
MAINS POWER PACK FOR MINIAMP 2	**MPP1 £5.99**
LIGHTWEIGHT QUALITY STEREO HEADPHONES	**SHP1 £5.99**

NEW — AMIGA MIDI INTERFACES

MIDI INTERFACE 1 FOR A500/2000 OR A1000 .... **MIF1 £29.99**
This has MIDI IN, OUT & THRU sockets. Self powered.
MIDI INTERFACE 2 FOR A500/2000 OR A1000 .... **MIF2 £34.99**
This has MIDI IN, MIDI OUT, MIDI THRU plus 2 UNIQUE switchable MIDI OUT/THRU SOCKETS to save you having to swap MIDI leads when using more than one MIDI instrument. Self powered.

SUNDRY ITEMS — BARGAIN PRICES

SONY 3.5" DISKS 100% Guaranteed DS/DD disks

Lowest Prices ... Per 10 **£12.99 & Postage** Lowest Prices ....Per 25 **£28.49 Post Free**
Lowest Prices .... Per 50 **£55.99 Post Free** Lowest Prices ....Per 100 **£99.99 Post Free**

3½ " DISK BOX FOR 50 DISKS. SMOKED LID, LOCKABLE, DSB4	**ONLY £8.45**
3½ " DISK BOX FOR 100 DISKS. SMOKED LID, LOCKABLE, DSB6	**ONLY £12.49**
A500 DUSTCOVER. ANTISTATIC TYPE, ADC1	**ONLY £4.99**
A4 NON-SLIP MOUSEPAD, AFM1	**ONLY £4.99**

WE ALSO STOCK AMIGAS, PCs, PRINTERS, MONITORS, HARD DISKS, MEMORY EXPANSIONS, VIDEO DIGITISERS, SELECTED SOFTWARE & BOOKS — PHONE FOR OUR FREE 20 PAGE CATALOGUE. TRADE ENQUIRIES WELCOME. Please add 75p part postage + packing to orders under £15.00

FREE CATALOGUE WITH ALL ORDERS *EXPRESS DELIVERY £1.99 Extra*

TRILOGIC

TRILOGIC, Dept. A.C. Unit 1, 253 New Works Road, Bradford BD12 0QP
Tel 0274-691115 Fax 0274 600150

- FAST MAIL ORDER SERVICE - PROMPT DESPATCH - ALL PRICES FULLY INCLUSIVE PLUS 10 DAY MONEY BACK GUARANTEE ON ALL ITEMS. PAYMENT BY CASH, CHEQUES PAYABLE TO TRILOGIC BY POSTAL ORDER, ACCESS OR VISA. ADD £2·99 EXTRA FOR EXPORT ORDERS, PAYMENT IN STERLING ONLY PLEASE.

Mailbag

Your chance to air your views in Your Commodore

Having very recently acquired an Oceanic 118N, I was particularly interested to read Norman Doyle's articles in various editions of *Your Commodore.* A comment made in that article has me a little worried. This concerns the remarks made about flippy disks in the paragraphs on the Commodore 1570/71. I wonder if the friction damage caused by turning disks over to use 'side 2' is a problem with the drives – could it be harmful to my OC-118N. If so then there is no economy in buying double sided disks.

I would be grateful for your comments on this.
F D A Rocha, London

I found the article 'Which Drive' by Norman Doyle in the June '88 issue of *Your Commodore* most interesting. With reference to his comments on the 1551:

I have Commodore models C64 with 1541 Disk Drive and a Plus/4 with 1551 Disk Drive. Both the Plus/4 and C16 will accept either type of disk drive since they are fitted with the usual 6 – pin DIN sockets for serial daisy chaining of peripherals and a User Port having a 2×25 contacts interface. This User Port is NOT the same as that of the C64. The 1551 drive has a fitted data transfer lead which terminates in a custom made cartridge to fit into the User Port. This cartridge has a 'through' connection in which to daisychain other cartridges, although I have never seen or heard of any. Obviously more 1551s could be used, but I merely plug the 1541 into the alternative serial socket when I need two drives running. Visually and physically the 1551 is similar to the 1541 except there is no provision for daisychaining a 6 pin DIN-connected printer, but this is unnecessary in view of the facility being provided on the Plus/4 and C16. The 'latch' is of the 'turn down tab' similar to the Excelerator and Blue Chip drives.

Performance comparison was carried out by entering a 34 block (actually 8,515 bytes) data-storage program from the 1551 drive into the Plus/4. A timing program – see below- was entered into the C64 and used to measure the time for the Plus/4 to load the 34 block program. Loading time was consistently nine seconds flat. The disk was then transferred to the 1541 for loading into the C64 and the timing program entered into the Plus/4. The tests were repeated on the 1541 and loading took 23 seconds. Using these figures: 1551 transfers at 9460 baud compared with 1541 transferring at 3702 baud.

This clearly indicates an improvement of nearly 250% when loading program files via the 1551. Speeds may well differ on other disk operations or filing systems.

The Commodore 1551 disk was distributed in some quantity, but rarely, if ever, advertised. I bought mine from Green's in Nottingham's branch of Debenhams late in 1986 for about £159. They had several dozen for sale at the time. The odd one has been offered in FOR SALE columns of various publications.

Although games are not my forte one must accept that their wide spread appeal, especially to the younger members of society, generates the need for advertising, and is a necessary source of income for journals and magazines. Notwithstanding I would suggest that the more mature enthusiasts will continue to subscribe for many years if more articles in similar vein to Mr. Doyle's are forthcoming.

```
* 1 rem time program
2 TI$ = "000000"
3 GETA$: IFA$ = ""THEN 3
4 PRINT TI$: rem prints time when
5 RUN : rem any key pressed
```

Michael Stock, Nottingham

I made a mistake! I sold my 1541 and bought an excelerator!

Oh, don't get me wrong, it looks good, sounded good and loaded all my games, until I tried to load *Defender of the Crown.* Then the cursing began, crash after crash after crash. Instinct told me that the Excelerator was at fault because the game had always loaded on my 1541 with no aggro.

I rang Evesham Micros to explain the problems and was informed that, "Yes, *Defender of the Crown* would not load on the Excelerator drive" – for a moment I was lost for words! The dry voice on the phone then added: "the same applies to *Flying Shark*, they are the only two".

Apparently, Evesham have had to give away at least one Amiga computer because of *Defender of the Crown,* Although the manager quickly pointed out how big hearted they had been because the game did tend to mis-load occasionally, even on the 1541. He went on to suggest the fault lay with the software company, saying that they were at a loss as to the reason for this.

I then stated that I was very unhappy with the situation and was told that if I sent my copy of *Defender* to them (Evesham) they would re-configure the software to run on the Excelerator. The chap did explain that the Excelerator could not be made totally compatible with the Commodore because copyright laws would be infringed.
Alan Piela, Chingford, Essex

Following Norman Doyle's article on disk drives in the February issue, the letters have been flooding in. Now's your chance to set a few things straight.

Using the reverse side of a disk known affectionately as a 'Flippy' disk, is an established practise, born out of the need to cut costs. Now that floppy disks are cheaply available to the general public, the drawbacks become more obvious:

Because the disk is made to spin backwards when you turn it over to use the other side, the increased friction (you can hear the difference) must reduce the rotation speed, but so slightly as not to affect a healthy drive. The main drawback with flippys, is that spinning them backwards releases all the dust conveniently trapped inside the disk's outer sleeve. Although this is not a major menace, it tends to make the flippy side less reliable. I only use flippys for archival storage, just to save space in my disk box.

The question of drive compatibility has reared its ugly head once or twice already. It's all down to interpretation: if a manufacturer claims his product is 'Commodore compatible', it should be just that! There's no way anyone can make a truly compatible drive, that would involve replicating much that is protected by copyright laws! Competition, however, can't be a bad thing in this case, the customer will benefit from having a wider choice, after all! Bear in mind also, Commodore's history of incompatibility! The 1541 was not totally compatible with the older 1540 drive, the new 1541C is definitely not compatible with the 1541. Even C64's themselves have undergone design changes that have caused compatibility problems! You can't stop progress though, and as long as the software publishers keep abreast of these new developments, everybody should be happy!

Evesham Micros, distributor for the Excelerator drive, ended up giving away two Amigas, and their offer of a conversion job on your unloadable games still stands. They are quick to point out that that *Defender Of The Crown* originally would load on the 1541C. The situation was reversed when the game was modified! Evesham also point out that the Excelerator loads more games than Commodore's own 1541C! Ya pays ya money and ya takes ya choice!

Owing to the lack of popularity of the Plus/4 and C16 computers, Commodore decided to abandon the 1551 drive some time ago. Pity!

If you want to write to us, address your letters to:

Your Commodore,
Argus House,
Boundary Way,
Hemel Hempstead,
Herts HP2 7ST.

CLASSIFIED ADVERTISEMENT COUPON

YOUR COMMODORE

Write your advert in BLOCK CAPITALS in the grid below, ticking the section you wish it to appear under, INCLUDING NAME AND ADDRESS IN THE WORD COUNT and send it to: **YOUR COMMODORE, CLASSIFIED DEPARTMENT, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HP2 7ST.**
Lineage 53p per word (+ VAT). Minimum charge £7.95. Display box rate £11.50 per single column centimetre + VAT. Minimum size 2.0cm, no reimbursements for cancellations. All ads must be pre-paid.

☐ **For Sale** ☐ **Software** ☐ **Hardware** ☐ **Languages** ☐ **Services** ☐ **Amiga**

CLASSIFIED COUPON

ALL CLASSIFIED ADVERTISEMENTS MUST BE PRE-PAID. THERE ARE NO REIMBURSEMENTS FOR CANCELLATIONS.

I enclose my Cheque/Postal Order* for £........... for........... insertions, made payable to Argus Specialist Publications. (*Delete as necessary) or

PLEASE DEBIT MY ACCESS/BARCLAYCARD NO.

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiry Date ....

£................for.......insertions

Name ....

Address ....

....

.... Post Code ....

Day-time Telephone No: ....

Signature .... Date ....

YOUR COMMODORE

Lineage: 53p per word.
(+ VAT)
Semi display: £11.50 plus VAT per single column centimetre minimum 2cm. Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

0442 66650

Send your requirements to:
CLASSIFIED DEPARTMENT
ASP LTD. ARGUS HOUSE,
BOUNDARY WAY,
HEMEL HEMPSTEAD HP2 7ST.

BULLETIN BOARD

TBBCS Bulletin Board

Become a sysop for less

If you have an Amiga with two drives and a Hayes compatible modem, you could be running your own bulletin board for only 35 pounds!

TBBCS will let you build up a BBS with messages, voting, a dating service, SIGs, downloads, and a CB Simulator for less than the price of most Comms Packages. If you don't have your own phone line, TBBCS can share a voice line.

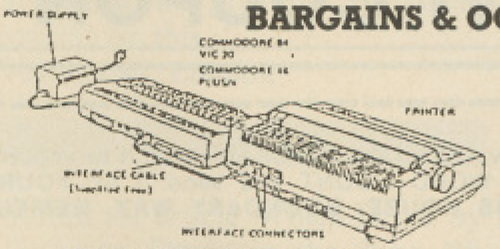
Send an SAE to the following address for more information.

Hugh Allan Jr, Ness Castle, Dores Rd, Inverness IV1 2DJ

★ **Stop ringing other peoples bulletin boards, and start running your own.** ★

HARDWARE

EEC LTD. CBM PRINTER BARGAINS & OCP ART STUDIO



★SEIKOSHA DOT MATRIX £125.00★
GP 500 VC c.p.s. Pin feed with CBM Serial connector for Commodores (as shown above for HRIC)

★CBM, OCP★
ART STUDIO £12.95 ADVANCED VERSION £19.95
a further £2 off if ordered with printer

★SPECIAL OFFER COMMODORE 64
P.S.U. £22 inc post ★

All prices include VAT Add £6 postage for printers. Other items £2. Tel orders VISA & ACCESS or cash with order. Ansaphone orders ANYTIME Please leave telephone number for acknowledgement.

★ BROTHER HR10C★
★ £129.95 ★

daisy wheel 60 col
12 CPS 2K Buffer,
FEED • BOOKS IN ENGLISH, GERMAN & FRENCH

IDEAL FOR WORD PROCESSING

Previously list price £343.00

18-21 Misbourne House,
Chiltern Hill,
Chalfont St Peter, Bucks.
SL9 9UE.
Tel: 0753 888866

HARDWARE

COTSWOLD COMPUTERS
MILTON KEYNES

Call today for the best prices in the Area.

AMIGA A500 inc MODULATOR
£374

AMIGA A500 + 10 Star Games Pack
£389

ST EXPLORER PACK
£299

Plus all software at discount prices

All prices include V.A.T.

22 Polruan Place, Fisher Mead, Milton Keynes MK6 2LH
Phone (0908) 662331

SERVICES

COMMODORE SUPPLIES

C64 Power Supply	£19.95
CBM Plus 4 Power Supply	£29.95
C-16 Power Supply	£12.95
1530 Datasette	£27.95
1531 datasette	£27.95
Simons Basic Extension	£9.95
Serial Lead (for d/drive/printer)	£5.95
C64 reset Switch	£4.95
Dust Cover (please state computer)	£3.50

Prices include VAT & P&P. Allow up to 14 days for delivery. Chq/PO to:
Omnidale Ltd (Y.C.) 23 Curzon Street, Derby, DE1 2ES. Tel 0332 291219.

SOFTWARE

ONLY POOLS AND HORSES

Two programs for the price of one
BOXFORM enables anyone to quickly weigh up a race. Track characteristics built in. Easily amended to suit the user's ideas.

HANDICAP WINNER shows the value for money in handicaps. Over 400 winners in 1989 include 25/1, 16/1, 2 at 14/1, 5 at 12/1, 4 at 11/1, 5 at 10/1, 9 at 9/1, 14 at 8/1, 10 at 7/1, 11 at 13/2, 18 at 6/1, 16 at 11/2, 23 at 5/1 etc. etc.

Tape £11.95, Disc £12.95 for C64/128
Other programs available covering all aspects of racing and football form.
Write or phone for details.

BOXOFT programs **SAVE TIME** not waste it **BOX**oft **CLEVER** and get the **BEST**

BOXOFT, YC, 65 Allans Meadow, Neston South Wirral. L64 9SQ
051-336-2668

SAMDALE COMPUTERS LTD

"Better than the best of the rest"

RING NOW Credit Card Hotline 0621 742617

THE LATEST SOFTWARE FOR COMMODORE C64/C128/AMIGA

Ring now for a free price list on 0621 742617.
or send stating name and address and machine type.

SAMDALE COMPUTERS LTD
1a Austral Way, Highfield Rise, Althorne, Essex CM3 6DN.

SPECIAL OFFERS

"You better believe OUR PRICES"

Amiga 500 + 10 Star games + gifts	£389.00
Commodore 1084S Colour Monitor	£240.00
C64 Compatible Data Recorder	£24.50
Commodore 1351 C64/C128 Mouse	£19.99
C64 Compatible Power Supply	£19.99
C64 Slimline Replacement Case	£20.00
Commodore 1541 C64 5¹/₄" D/Drive	£159.95
Oceanic 5¹/₄" D/Drive + FREE S/Ware	£119.95
3.5" Disks x 10	£9.50
5.25" Disks x 10	£4.50
Disk Box 80 x 3.5" (Lockable)	£6.99
Disk Box 100 x 5.25" (Lockable)	£6.99

ALL PRICES INCLUDE VAT
P/P. £1.50 SMALL ITEMS, £3.50 FOR HARDWARE

Crofton Micro Supplies
FREEPOST, London. SE4 2BR.
Telephone 01-469-3246

WANTED

WANTED for the C+4, Printer and Disk Drive. In good working order. Phone (0222) 382640.

WANTED information ICL per Q Computer Nigel. (0293) 513354.

YOUR COMMODORE SPECIALS

TECH DRAW 64 - A comprehensive technical illustration aid for C64 owners for both disk and tape. Commands available include LINE DRAW, RAYS, BOX, CIRCLE, ARC, ELLIPSE, FREEHAND, FILL, HATCH & TEXT. Wipe copy, move and rotate, save and load or print your drawing all for £8.95 tape or disk. Order Codes TD64T (tape) TD64D (disk)

Full details in Your Commodore July 1987.

SPEEDY ASSEMBLER

Your Commodore's very own Assembler, a 100% memory resident program for loading from tape or disk, featured in the Your Commodore Machine Code Series and now the Y.C. standard assembler only £9.95 Tape or Disk. Order Codes YSPASSD (disk) YSPASSR (tape)

Order from Your Commodore Readers Services at 9, Hall Road, Maylands Wood Estate, Hemel, Hempstead Herts HP2 7BH. Please make cheques payable to A.S.P. Ltd., or telephone your Access/Visa order on 0442 41221

PUBLIC DOMAIN SOFTWARE for the C64/128.

We have 150 disks full of all types of Programs
All disks are £2.95 each
NOW FORTH +4
from £3.75 per disk.
Send SAE or phone for free catalogue (state model)

Kingsway Computer Services
Dept 15 140 Rushdale Road, Sheffield S8 9QE
Tel: (0742) 588429

FOR SALE

esoteric

DISK MATE	1541/C64 Disk Toolkit
SECUREWORD	WP Data Encryption
PSYCHOLOGY +	Analyse your character.

Disk Only. **£8.00** inc p&p

15 Lanmoor Est. Lanner, Redruth, Cornwall. TR16 6HN

B R E A K

Readers Problems

Though the Commodore 64 is one of the world's most popular microcomputers, it can be very difficult to find specific information about your particular machine.

At the *Your Commodore* office we receive literally hundreds of letters from you, our readers, on a wide range of subjects ranging from the simple 'Can you give me the telephone number for...', to the more complex 'I'm trying to write a program that uses a split screen. How do I do it?'

Unfortunately, the volume of mail received has become so great that it is impossible to answer every letter and still manage to publish a magazine each month.

For this reason we have felt it necessary to produce a number of guidelines for getting information from us:

1) We cannot guarantee to answer every letter sent to the magazine. Should it become apparent that a number of readers are suffering from the same problem, then we will reply to the letter via the Letters page.

2) A new helpline has been set up. This will be open for your queries on Tuesday and Thursday afternoons between 2.00pm and 4.00pm. We will not be able to deal with our telephone queries at any other time. If our technical adviser is not available when you ring, then a message will be taken.

3) If you are having problems with one of our listings, can you please let us know in writing. This will enable us to see if a number of people are having the same problem. When a common problem becomes apparent with a program, then a correction sheet will be issued. Enclose a self-addressed, stamped envelope and we will send you a copy of the correction sheet as soon as it is available.

We are sorry that it has become necessary to instigate these rules. However, we are sure that you will agree with us that the more time that we can spend making *Your Commodore* the most informative magazine around, the better.

For program queries write to:
Program Corrections
Your Commodore
Argus House,
Boundary Way,
Hemel Hempstead
HP2 7ST
Tel: 0442-66551

Oops!

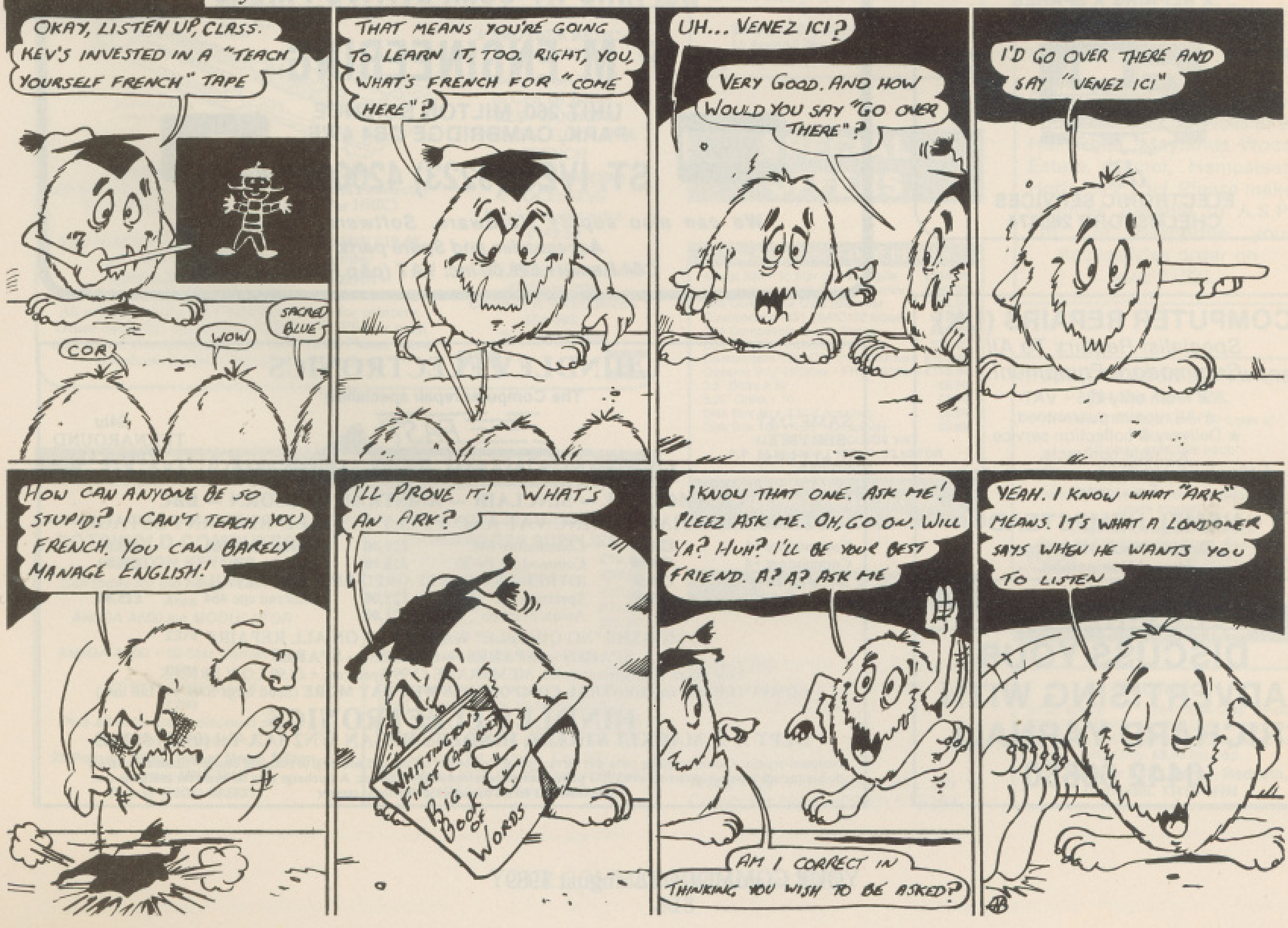
Unfortunately, in the July issue, we missed part of the listing for the Windows Demo program. The missing lines are reprinted here.

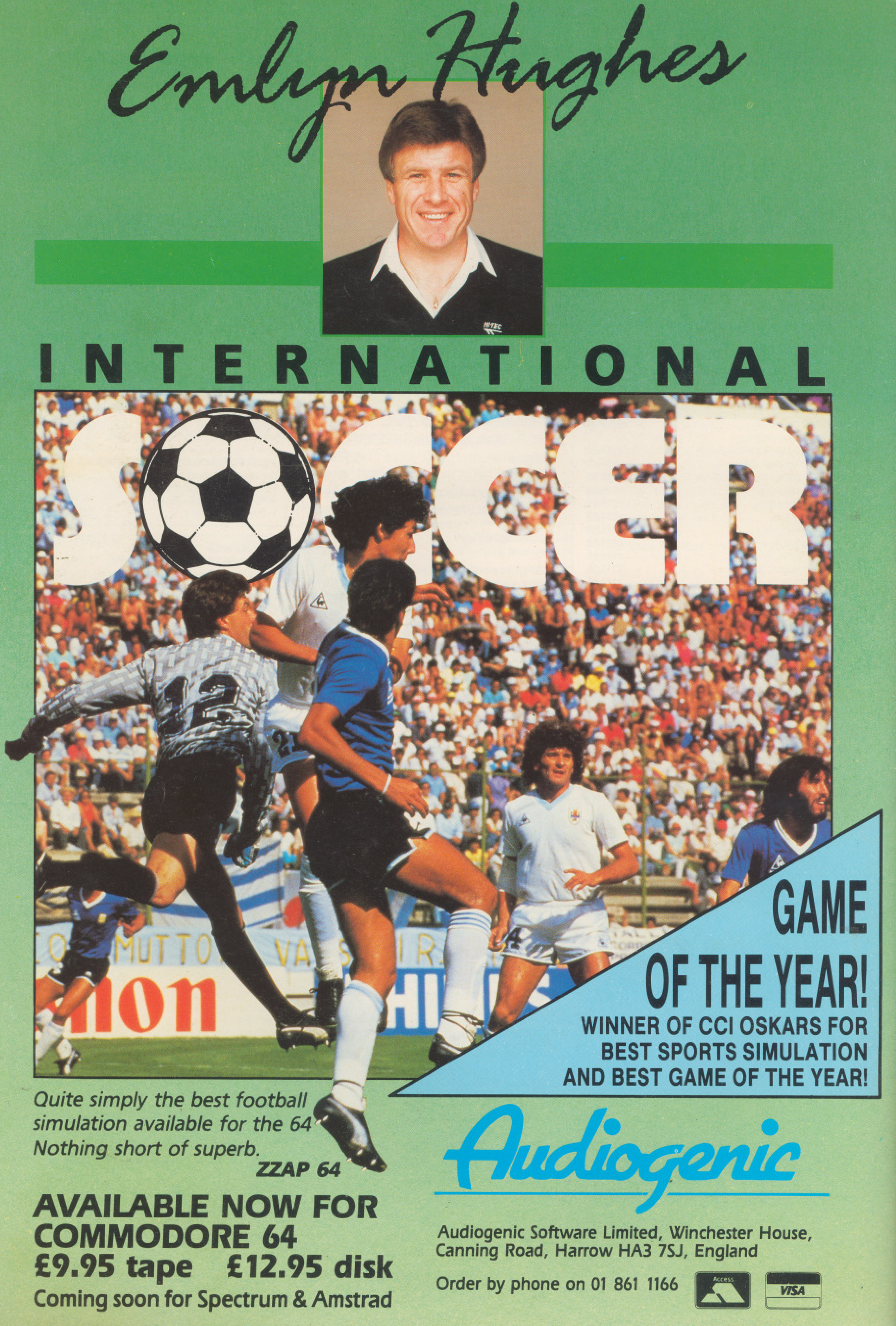
```
FO  790 A=ASC(A$)
F1  800 IFA=138THENSYSUP:GOTO740

AE  810 IFA=134THENSYSDOWN:GOTO7
        40
18  820 IFA=139THENSYSLEFT:GOTO7
        40
59  830 IFA=135THENSYSRIGHT:GOTO
        740
C5  840 RETURN
DA  850 REM
CB  860 REM     PRINT DISK DIRECT
        ORY
E6  870 REM
OE  880 OPEN2,8,15:Z=0:P=1:PRINT
        #2,"I
3F  890 OPEN1,8,0,"$0"
43  900 GET#1,A$,B$
05  910 GET#1,A$,B$
37  920 GET#1,A$,B$
A8  930 C=0
2B  940 IF A$<>"" THEN C=ASC(A$)

9B  950 IF B$<>"" THEN C=C+ASC(B
        $)*256
```

The Nibbles *By Alan Batchelor*




Emlyn Hughes
INTERNATIONAL
SOCCER
GAME
OF THE YEAR!
WINNER OF CCI OSKARS FOR
BEST SPORTS SIMULATION
AND BEST GAME OF THE YEAR!
Quite simply the best football
simulation available for the 64
Nothing short of superb.
ZZAP 64
Audiogenic
AVAILABLE NOW FOR
COMMODORE 64
£9.95 tape £12.95 disk
Coming soon for Spectrum & Amstrad
Audiogenic Software Limited, Winchester House,
Canning Road, Harrow HA3 7SJ, England
Order by phone on 01 861 1166
VISA